HYDRO DATA CART WEB SERVICE API

Service End Point: http://myserver/hydroCart

Development Stages:

1) No saving
2) Saving only one cart per user
3) Named carts

Web Service API:

1. **Empty a data cart:** remove all contents of a cart.

   Request:

```
cmd=empty&sessionid=xxxx&username=yyyyy&cartname=zzzzz
```

- username will be encrypted based on certain seeds. This is for security purpose.
- If username is not specified, “anonymous” will be used. If cartname is not specified, “default” will be used. With these default values, the same code can be re-used for stages in which we support saving data carts and support saving named data carts.

   Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<response>

        <status>SUCCEED|FAIL</status>

        <!-- in case fail, error message is included in info element below

Otherwise it may be a message like:

Cart for user UUUUU accepted with NNN items-->

        <errorinfo>error message</errorinfo>

</response>
```

2. **Save to a data cart: This will save user's selections to a data cart.**

   Request:

   cmd=save&sessionid=xxxx&username=yyyyy&cartname=zzzzz&xml={xml}

   - Notes for username and cartname: see 1.
   - XML can be very long depending on the number of sites the user selected. HTTP POST will be used to pass the request.
   - serviceCode should
   - The format of the xml is as the following:

<?xml version="1.0" encoding="UTF-8"?>

```
<datacart >
<!-- describe the cart
When creating, cartname is required. If email is provided, just send link to
the person
When read is used, return the date the cart was created, and when it may be
removed.
-->
  <cartDescription>
    <cartname>HydrocartExample</cartname>
<!--  At creation time only. If email is provided, email user when data is
ready -->
    <emailDownloadLinkTo>string</emailDownloadLinkTo>

<!-- When read is used, return the date the cart was created, and when it may
be removed.
-->
    <cartCreated>1984-08-20T11:30:43.46</cartCreated>
    <deletionDate>1976-02-28</deletionDate>
-->
  </cartDescription>

<!-- suggested list of services that information is to be retrieved from
Hints, while default service locations may be stored with the data cart
service. There will be times we want to use a service that is unknown. To the
service.
-->
  <cuahsiServices>

<!--
A waterML service
Required elements:
   serviceCode is used as the reference
   ServiceWSDL can be used to read the SOAP endpoint.
Optional elements:
  serviceVersion  version of the service
  serviceEndpoint  url of the endpoint (needed? Discoverable )
  serviceName   Name of the service that might be rendered in html.
```

Formatted: Font: Bold

Formatted: Pattern: Clear (Custom Color(RGB(242,242,242)))

Formatted: Pattern: Clear (Custom Color(RGB(242,242,242)))

```
-->
    <waterMLService>
      <serviceCode>NWISIID</serviceCode>
      <serviceWsdl> http://river.sdsc.edu/wateroneflow/NWIS/Data.asmx?WSDL
            </serviceWsdl>
<!-- optional -->
            <serviceVersion>1.0</serviceVersion>
<!-- optional. Can be discovered using serviceWsdl. -->

      <serviceEndpoint>http://river.sdsc.edu/wateroneflow/NWIS/Data.asmx
</serviceEndpoint>
<!-- optional: name to
            <serviceName> USGS NWIS Instantaneous Irregular data
</serviceName>
</waterMLService>
<!-- Basic required elements to describe a service -->
    <waterMLService>
      <serviceCode>NWISDV</serviceCode>
      <serviceWsdl>
http://river.sdsc.edu/wateroneflow/NWIS/DailyValues.asmx?WSDL</serviceWsdl>
    </waterMLService>

  </cuahsiServices>


<!-- Data requests
The meat of the data cart.
Contains a list of GetValuesRequest elements, and it's parameters, plus
optional description

-->
  <serviceRequests>
      <!-- GetValuesRequest
       @serviceCode, match a known one, or one from above in the
      cuahsiServices
       elements-->
    <GetValuesRequest serviceCode="NWISIID">
      <location>NWISIID:324444114385901</location>
      <variable>NWISIID:00410</variable>
      <startDate>1964-04-30</startDate>
      <endDate>1964-05-30</endDate>
      <!-- optional. Hint to render site name in client -->
      <siteName>016S022E21R001S</siteName>
      <!-- optional. Hint to render variable name in client -->
      <variableName> Acid neutralizing capacity, water, unfiltered, fixed endpoint (pH 4.5) titration,
field, milligrams per liter as calcium carbonate  </variableName>
    </GetValuesRequest>

    <GetValuesRequest serviceCode="NWISDV">
      <location>NWISDV:10263500</location>
      <variable>NWISDV 00060</variable>
      <startDate> 2005-08-01</startDate>
      <endDate> 2006-08-01</endDate>
<!-- No siteName or variableName elements -->
    </GetValuesRequest>
```

```
    <GetValuesRequest serviceCode="NWISDV">
      <location>NWISDV: 363143110355001</location>
      <variable>NWISDV:72019</variable>
      <startDate> 2006-08-01</startDate>
      <endDate> 2006-08-16</endDate>
    </GetValuesRequest>

    <GetValuesRequest serviceCode="NWISUV">
      <location> NWISUV:10109000</location>
      <variable>NWISUV:72019</variable>
<!-- Moving Real time window. Just ask for all data -->
    </GetValuesRequest>

  </serviceRequests>
</datacart>
```

3. **Read a data cart**: This will return the contents in a cart, thus browser side code (e.g. JavaScript) can display the contents in the cart.

   Request:

   cmd=read&sessionid=xxxx&username=yyyyy&cartname=zzzzz

   - Notes for username and cartname: see 1.

Example of <queryInfo> element

```
<queryInfo>

 <creationTime>2008-08-20T18:40:39.762375Z</creationTime>

 <criteria>

  <locationParam>LittleBearRiver:USU-LBR-Mendon</locationParam>

  <variableParam>LBR:USU10</variableParam>

  <timeParam>

   <beginDateTime>2007-08-01T00:00:00</beginDateTime>

   <endDateTime>2007-12-31T00:00:00</endDateTime>

  </timeParam>

 </criteria>
```

Comment [tw1]: Dean, I see this first iteration of "data cart" as supporting one purpose: downloading time series data (i.e., calling GetValues). As such, I would structure the response to support this, which means having the same types of parameters as are in the GetValues call, and having all the information (such as the WSDL URL) required for a client to make a call to the service. Most of this information is already defined in the <queryInfo> element of WaterML. Just add the URL, and you are good to go. I've copied and pasted an example of the element, as retrieved by calling GetValues the a site in the Little Bear River network. This element is included in the WaterML returned from a call to GetValues, as a record of the query that the client made.

Comment [tw2]: The parts to pay attention to in here are all within the <criteria> element.

Formatted: Font: 13 pt, Bold

Comment [tw3]: Maybe before the location param, you could have a <serviceURL> element that has the URL for the WSDL of the web service, e.g.:

<serviceURL>http://his02.usu.edu/littlebearriver/cuahsi_1_0.asmx?WSDL</serviceURL>

<note>OD Web Service</note>

</queryInfo>

Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<response>

    <status>SUCCEED|FAIL</status>

        <!-- in case fail, error message is included in info element below

    Otherwise it may be a message like:

    Cart_data is ready. Call Get

    -->

        <info>error message</info>

    <!-- in case fail, error message is included below -->

    <error>error message</error>
```

```
<datacart >
  <cartDescription>
    <cartname>HydrocartExampleShort</cartname>

<!-- When rRead was called, return the date the cart was created, and when it
may be removed. -->
    <cartCreated>2008-08-28T11:30:43.46</cartCreated>
    <deletionDate>2008-09-28</deletionDate>
  </cartDescription>

  <cuahsiServices>

    <waterMLService>
      <serviceCode>NWISIID</serviceCode>
      <serviceWsdl> http://river.sdsc.edu/wateroneflow/NWIS/Data.asmx?WSDL
     </serviceWsdl>
      </waterMLService>

    <waterMLService>
      <serviceCode>NWISDV</serviceCode>
      <serviceWsdl> http://river.sdsc.edu/wateroneflow/NWIS/DailyValues.asmx?WSDL
            </serviceWsdl>
    </waterMLService>

  </cuahsiServices>


  <serviceRequests>
    <GetValuesRequest serviceCode="NWISIID">
```

Formatted: Font color: Black

```
      <location>NWISIID:324444114385901</location>
      <variable>NWISIID:00410</variable>
      <startDate>1964-04-30</startDate>
      <endDate>1964-05-30</endDate>
      <!-- optional. Hint to render site name in client -->
      <siteName>016S022E21R001S</siteName>
      <!-- optional. Hint to render variable name in client -->
      <variableName> Acid neutralizing capacity, water, unfiltered, fixed endpoint (pH 4.5) titration,
field, milligrams per liter as calcium carbonate  </variableName>
    </GetValuesRequest>

    <GetValuesRequest serviceCode="NWISDV">
      <location>NWISDV:10263500</location>
      <variable>NWISDV 00060</variable>
      <startDate> 2005-08-01</startDate>
      <endDate> 2006-08-01</endDate>
    </GetValuesRequest>


  </serviceRequests>
</datacart>
```

```
</response>
```

4. **Get a data cart:** This will return a URL, pointing to a zipped Access database containing the metadata (, which doesn't contain the time series data) of cart.

   Request:

```
cmd=get&sessionid=xxxx&username=yyyyy&cartname=zzzzz
```

   - Notes for username and cartname: see 1.

   Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<response>

    <status>SUCCEED|FAIL</status>
```

```
    <!-- in case fail, error message is included in info element below

    Otherwise it may be a message like:

    Zip Download size is xxx K-->

            <info>error message</info>
```

```
    <!-- in case fail, error message is included below -->

    <error>error message</error>
```

```
    <output>http://myserver/output/20080817120203242.zip</output>

</response>
```

5. **Download a data cart:** This will extract time series data of each site selected, convert them into a personal geodatabase, zip it to a certain web accessable folder, send the URL in an email to the user, and return the URL as well.

Request:

```
cmd=download&sessionid=xxxx&username=yyyyy&cartname=zzzzz&email=mmm
```

Notes:

- username, cartname, and email address will all be encrypted for security purpose.
- If no email address is specified, no email notification will be sent.

Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<response>

    <status>SUCCEED|FAIL</status>

    <!-- in case fail, error message is included in info element below

    Otherwise it may be a message like:

    Zip Download size is xxx K-->

            <info>error message</info>

    <!-- in case fail, error message is included below -->

    <error>error message</error>

    <output>http://myserver/output/20080817120203242.zip</output>

</response>
```

Thoughts:

Error changed to info. Useful information can be communicated.

Use http status to communicate

Formatted: Heading 2, Indent: Left: 0", Pattern: Clear

- success (200 OK),
- Fail (400 Bad Request)

Designed for immediate response. Large lists of sites do not come back fast.

Possible XML schema.

Compile to code with xsd.exe

```
<?xml version="1.0" encoding="utf-8" ?>

<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="response" type="ResponseType" />
  <xs:element name="datacart" type="DataCartRequestType">
    <xs:annotation>
      <xs:documentation>datacart is submitted to create a data cart, and
returned when data cart is read.

It includes a cart description, a list of services, and a list of request to
those services.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="ResponseType">
    <xs:annotation>
      <xs:documentation>A response is the message returned
when</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="status"
type="StatusTypeCodeList">
        <xs:annotation>
          <xs:documentation>SUCCEED
FAIL</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="info">
        <xs:annotation>
          <xs:documentation>element that contains error information when
status is FAIL.
Informational messages, like size of file, number of sites when status is
SUCCEED.
</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string" />
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:choice minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>One of three elements.

SessionCode when cart is created
Dataset when cart is read
outputURL for get and download
</xs:documentation>
        </xs:annotation>
        <xs:element name="sessionCode" type="SessionCodeType">
          <xs:annotation>
```

```
            <xs:documentation>SessionCode returned when a cart is
created.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="datacart" type="DataCartRequestType">
          <xs:annotation>
            <xs:documentation>Datacart definition that was submitted when
cart was created.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ouputUrl" type="outputUrlType">
          <xs:annotation>
            <xs:documentation>url of file when get and download is
called.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="GetValuesRequestType">
    <xs:annotation>
      <xs:documentation>A GetValuesRequestType contains information on how to
search for site-variable-time interval

GetValues Method has Four Parameters,

Dates can be null

GetValues (location,variable,beginDate,endDate, "AuthToken")

ODM:USU-LBR-Mendon

usu:999004/valueType=Field Observation/DataType=Average

2006-08-01

2006-08-15</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="location" type="xs:string">
        <xs:annotation>
          <xs:documentation>parameter that is submitted as location to
GetValues

network:siteCode
ODM:USU-LBR-Mendon

For some services network is overloaded:
 GEOM:Point(Lat Lon)
 GEOM:BOX(s w, n e)</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="variable" type="xs:string">
        <xs:annotation>
          <xs:documentation>variable:
usually:
"vocabulary:vocabularyCode"
```

```
eg.
usu:USU-LBR-Mendon

More specific for service that support options:
"VOCABULARY:code/option/optionN"

eg.
usu:999004/valueType=FieldObservation/DataType=Average</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="startDate"
type="xs:string">
        <xs:annotation>
          <xs:documentation>begin date for data retrieval:
2006-08-01</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="endDate"
type="xs:string">
        <xs:annotation>
          <xs:documentation>end date for data retireval
2006-08-01
</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="siteName"
type="xs:string">
        <xs:annotation>
          <xs:documentation>optional site name that can be used to renderhuan
readable descriptionsin a client.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="variableName"
type="xs:string">
        <xs:annotation>
          <xs:documentation>optional variable name that can be used to
renderhuan readable descriptionsin a client.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="serviceCode" type="xs:NMTOKEN" use="required">
      <xs:annotation>
        <xs:documentation>REQUIRED:
The serviceCode matching a serviceCode listed aboved in the cuahsiServices

</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="CuahsiServicesType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="waterMLService"
type="WaterMlServiceType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="outputUrlType">
    <xs:simpleContent>
```

```
      <xs:extension base="xs:anyURI" />
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="DataCartRequestType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="cartDescription"
type="CartDescriptionType">
        <xs:annotation>
          <xs:documentation>information to describe the cart, name, when it
was created, and scheduled to be deleted.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="cuahsiServices"
type="CuahsiServicesType">
        <xs:annotation>
          <xs:documentation>list of web services that the serviceRequests
will use.
Centralizing avoid having the same URL repeated in many requests. If this url
was to change... this means it only has to change in one
location.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="1" maxOccurs="1" name="serviceRequests">
        <xs:annotation>
          <xs:documentation>List of serviceRequest, eg GetValues method
calls.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:choice>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="GetValuesRequest" type="GetValuesRequestType">
              <xs:annotation>
                <xs:documentation>service requests contains zero or more
GetValuesRequests.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="WaterMlServiceType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="serviceCode"
type="xs:normalizedString" />
      <xs:element minOccurs="1" maxOccurs="1" name="serviceWsdl"
type="xs:anyURI" />
      <xs:element minOccurs="0" maxOccurs="1" name="serviceVersion"
type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="serviceEndpoint">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:anyURI">
              <xs:attribute name="protocol" type="xs:string" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
```

```
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="serviceName"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CartDescriptionType">
    <xs:annotation>
      <xs:documentation>This contains the description of a cart.

cartname is requied, and should not have any spaces

Include emailDownloadLinkTo, only if the cart is to be downloaded later.

This will allow for the description of a cart to be
expanded.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="cartname"
type="xs:normalizedString">
        <xs:annotation>
          <xs:documentation>cartname

When creating, cartname is required.


When read is used, return the date the cart was created, and when it may be
removed.

</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="emailDownloadLinkTo">
        <xs:annotation>
          <xs:documentation>Contains the email address to send link when
request is complete.

 If email is provided, just send link to the person. otherwise user is
expecte to do thier own get request.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:normalizedString" />
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element minOccurs="1" maxOccurs="1" name="cartCreated"
type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>when cart was created</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" maxOccurs="1" name="deletionDate"
type="xs:date">
          <xs:annotation>
            <xs:documentation>when it is scheduled to be
deleted</xs:documentation>
```

```
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="StatusTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SUCCEED" />
      <xs:enumeration value="FAIL" />
      <xs:enumeration value="RUNNING" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="StatusTypeCodeList">
    <xs:union memberTypes="StatusTypeEnum xs:string" />
  </xs:simpleType>
  <xs:simpleType name="FormatTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="esri.file_geodatabase.archydro" />
      <xs:enumeration value="esri.personal_geodatabase.archydro" />
      <xs:enumeration value="cuahsi.waterML_1_0.xml" />
      <xs:enumeration value="cuahsi.excel_2003.xml" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="FormatTypeCodeList">
    <xs:union memberTypes="FormatTypeEnum xs:string" />
  </xs:simpleType>
  <xs:simpleType name="SessionCodeType">
    <xs:restriction base="xs:NMTOKEN" />
  </xs:simpleType>
</xs:schema>
```

Formatted: Indent: Left: 0", Pattern: Clear