CHyF REST Web Services, v1.1

Prepared by: Chris Hodgson, Refractions Research (chodgson@refractions.net)
Emily Gouge, Refractions Research (egouge@refractions.net)

Prepared for: Mark Sondheim, Natural Resources Canada (mark.sondheim@canada.ca)
Marie-Eve Martin, Natural Resources Canada (marie-eve.martin@canada.ca)
Mark Laguë, Natural Resources Canada (mark.lague@canada.ca)
Denis Boutin, Natural Resources Canada (denis.boutin@canada.ca)

Date: 2019 January 31

Table of Contents

1 Overview

The CHyF REST web service is primarily intended to support the web-based demo app. However, it has become apparent that the REST service may be of use outside of the scope of this project and thus its design may be of interest to others.

2 REST Service Structure

2.1 Content

The REST services will return a representation of the requested feature(s) in the requested content type. The content of a feature includes at a minimum the geometry and associated attribute values. An exception to this is when a list of features is returned, it may be of use to return only the list of ID values, rather than the complete content. A more verbose option would include links (in HTML) or IDs (in other formats) to enable traversal of the backing graph structure, e.g.:

- Lists of the immediate upstream and downstream connections.
- Link from elementary flowpath to containing elementary catchment and reverse.

A parameter could be used to specify what level of verbosity is requested; the *idOnly* parameter described below is a basic example of such a parameter.

Note that depending upon the service, what is returned may not have an ID. For example, the watershed determined as upstream of a given point or elementary flowpath may be represented as a collection of primary catchments or as a single catchment based on amalgamating the relevant elementary catchments.

The content would be output appropriately in the requested format, using available data structures, e.g. JSON format would use GeoJSON, HTML would display a table of attributes and values, etc.

2.2 Services

There will be an HTTP REST service endpoint such as:

`http://chyf.ca/chyf/`

Within this service endpoint, there would be various services returning elementary flowpaths, elementary catchments, and other features.

In the future we may wish to support multiple variants of the data (numbered versions or other named variations of the data). Such variants might have a service endpoint such as:

http://chyf.ca/chyf/<variant>/

The “variant” placeholder is intended to allow for a version number or other named “variation” of the CHyF dataset. Its purpose is to allow the existing and past URLs into the REST service to be accessible into the future even as the data is updated and ID values are changed. This would allow Linked Open Data or other links into the data to continue to be accessible; however to access new, updated data, such links would need to be re-built against the new data - which would be easily done using the REST services described here. Maintaining access to historic data through such URLs would require that the past data sets were also stored and made available through the service.

2.2.1 Nexus

Nexus features would be available under the service URL:

http://chyf.ca/chyf/nexus

Including the following features:

/nexus/<nexus-id>.<format>

Returns a single nexus feature based on its ID, which is provided as nexus-id.

/nexus/within.<format>?bbox=<minx>,<miny>,<maxx>,<maxy>

Returns a list of the nexuses intersecting a given spatial bounding box defined by bbox.

/nexus/near.<format>?point=<x>,<y>&maxDistance=<dist>&maxFeatures=<n>

Returns a list of the nexuses near the given point in increasing order of distance, up to the maximum number and/or distance requested.

2.2.2 Elementary Flowpath

Elementary flowpath features would be available under the service URL:

http://chyf.ca/chyf/eflowpath

Including the following services:

/eflowpath/<eflowpath-id>.<format>

Returns a single elementary flowpath feature based on its ID, which is provided as eflowpath-id.

/eflowpath/within.<format>?bbox=<minx>,<miny>,<maxx>,<maxy>

Returns a list of the elementary flowpaths intersecting a given spatial bounding box defined by bbox.

/eflowpath/near.<format>?point=<x>,<y>&maxDistance=<dist>&maxFeatures=<n>

Returns a list of the elementary flowpaths near the given point in increasing order of distance, up to the maximum number and/or distance requested.

/eflowpath/flowsFrom.<format>?point=<x>,<y>

Returns the elementary flowpath nearest the given point and associated with the same elementary catchment as the point.

/eflowpath/upstreamOf.<format>?point=<x>,<y>

Returns a list that includes the elementary flowpath that the given point would flow into, and all elementary flowpaths upstream of it.

/eflowpath/downstreamOf.<format>?point=<x>,<y>

Returns a list that includes the elementary flowpath that the given point would flow into, and all elementary flowpaths downstream of it.

/eflowpath/<eflowpath-id>/upstream.<format>?maxFeatures=<n>

Returns a list of the elementary flowpaths upstream of the specified elementary flowpath.

/eflowpath/<eflowpath-id>/downstream.<format>?maxFeatures=<n>

Returns a list of the elementary flowpaths downstream of the specified flowpath.

/eflowpath/filter.<format>?property=<propertyName>
&predicate=(equals|greaterThan|lessThan)&value=<value>

Returns a list of the elementary flowpaths which meet the filter criteria. The propertyName must match a property name in the “properties” object in the GeoJSON feature, e.g. name, nameId, type, rank, strahleror, hortonor, hackor, or length.

This is intended to provide a basic way to query a subset of the elementary flowpaths; more complicated filtering would likely be better handled using WFS and the OGC filter encoding standard.

2.2.3 Elementary Catchment

Elementary catchment features would be available under the service URL:

`http://chyf.ca/chyf/ecatchment`

Including the following services:

`/ecatchment/<ecatchment-id>.<format>`

Returns a single elementary catchment feature based on its ID, which is provided as ecatchment-id.

`/ecatchment/containing.<format>?point=<x>,<y>`

Returns the elementary catchment containing the specified point.

`/ecatchment/within.<format>?bbox=<minx>,<miny>,<maxx>,<maxy>`

Returns a list of the elementary catchments intersecting a given spatial bounding box defined by bbox.

`/ecatchment/near.<format>?point=<x>,<y>&maxDistance=<dist>&maxFeatures=<n>`

Returns a list of the elementary catchments near the given point in increasing order of distance, up to the maximum number and/or distance requested.

`/ecatchment/upstreamOf.<format>?point=<x>,<y>`

Returns a list including the elementary catchment containing the given point, and all elementary catchments upstream of it.

`/ecatchment/downstreamOf.<format>?point=<x>,<y>`

Returns a list including the elementary catchment containing the given point, and all elementary catchments downstream of it.

/ecatchment/<ecatchment-id>/upstream.<format>?maxFeatures=<n>

Returns a list including the specified elementary catchment and all elementary catchments upstream of it.

/ecatchment/<ecatchment-id>/downstream.<format>?maxFeatures=<n>

Returns a list including the specified elementary catchment and all elementary catchments downstream of it.

Note that filtering based on property values is not supported at this time for elementary catchments. It could be added in the future if warranted.

2.2.4 Catchment (Drainage Area)

Drainage Area features would be available under the service URL:

http://chyf.ca/chyf/drainageArea

Drainage area features are aggregates of the requested elementary catchments. As they are dynamically created features, they do not have an ID or other attributes.

This includes the following services:

/drainageArea/upstreamOf/ecatchment/<ecatchment-id>.<format>

Returns a feature representing the drainage area upstream of the specified elementary catchment.

/drainageArea/downstreamOf/ecatchment/<ecatchment-id>.<format>

Returns a feature representing the drainage area downstream of the specified elementary catchment.

/drainageArea/upstreamOf/eflowpath/<eflowpath-id>.<format>

Returns a feature representing the drainage area upstream of the specified elementary flowpath.

/drainageArea/downstreamOf/eflowpath/<eflowpath-id>.<format>

Returns a feature representing the drainage area downstream of the specified elementary flowpath.

/drainageArea/upstreamOf/location.<format>?point=<x>,<y>

Returns a feature representing the drainage area upstream of the specified point location.

/drainageArea/downstreamOf/location.<format>?point=<x>,<y>

Returns a feature representing the drainage area downstream of the specified point location.

All of the above services support an additional parameter “removeHoles”, which, if set to the value “true”, will return the drainageArea polygon with any interior holes removed. E.g.:

/drainageArea/upstreamOf/location.<format>?point=<x>,<y>&removeHoles=true

2.2.5 Waterbody Flows

This service is similar to the flowpath upstream/downstream from point service, except in cases where flowpaths are associated with polygonal waterbodies (e.g., a double-line river or a lake) where the polygonal waterbody is returned instead of the linear flowpath. The resulting layer contains both Linear and Polygonal features and the associated attribute table contains the attributes that are applicable to both the elementary flowpaths and the waterbody catchments.

http://chyf.ca/chyf/multiDimensionalUpstreamOf.<format>?point=<x>,<y>

Returns a list that includes the elementary flowpath and/or waterbody catchments that the given point would flow into and all elementary flowpaths and/or waterbody catchments upstream of it.

http://chyf.ca/chyf/multiDimensionalDownstreamOf.<format>?point=<x>,<y>

Returns a list that includes the elementary flowpath and/or waterbody catchments that the given point would flow into and all elementary flowpaths and/or waterbody catchments downstream of it.

2.2.6 Pourpoint Service

A pourpoint is an arbitrary location on a flowpath network, which may be of interest because of the existence of a water survey gauge, a sampling site, a confluence, or some other feature of hydrological and environmental interest. The pourpoint service accepts a list of pourpoints as input and returns a variety of outputs representing upstream catchments and their relationships.

```
http://chyf.ca/chyf/pourpoint/compute.<format>
      ?point=<id1>,<x1>,<y1>,<ccode1>,<id2>,<x2>,<y2>,<ccode2>...,
            <idN>,<xN>,<yN>,<ccodeN>
      &output=(op|opdmin|opdmax|pc|pcc|noc|nocr|tcc|tccr|ic),...
      &removeHoles=<true|false>&includeStats=<true|false>
```

For example:

```
http://chyf.ca/chyf/pourpoint/compute.json
      ?points=P1,-73.07844,45.45595,-1,P2,-73.07200,45.46100,-1
      &output=op,nocr,opdmin,opdmax,pc,pcc,noc,tcc,tccr,ic
      &removeHoles=true&includeStats=false
```

For each input pourpoint, four values must be provided in the “point” input parameter list: an external ID string (used as a reference in the results), an x and y coordinate pair, and a “ccode” value as specified in the first table below. The output parameter allows the user to specify which of the various possible outputs are desired, using a list of output key values, as defined in the second table below. Input coordinates are assumed to be in EPSG 4617 unless other specified using the srs parameter described in Section 2.3 Parameters.

Pourpoint C-Codes

Code	Description
-2	The closest hydro nexus contained by the same catchments the pourpoint lies within.
-1	**Not Supported in this Version.** The intention of this C-Code is to project the pourpoint to the nearest point on the nearest flowpath. The elementary flowpath and the corresponding elementary catchment will be split at this point and the upstream catchment computed from this break.
0	The closest hydro nexus to the pourpoint, with the results including all upstream catchments from this point.
>0	The closest hydro nexus to the pourpoint, with the results including only the Nth inflow catchment.

Output Options

Key	Name	Description
op	Output Pourpoints	The input pourpoints projected to the flow network as identified by their c-code.
opdmin	Pourpoint Minimum Distance Matrix	The minimum distance along the flow network between any two pourpoints (or null if the two pourpoints are not on the same flow network).
opdmax	Pourpoint Maximum Distance Matrix	The maximum distance along the flow network between any two pourpoints (or null if the two pourpoints are not on the same flow network).
opdprimary	Pourpoint Primary Distance Matrix	The distance along the flow network between any two pourpoints following primary flows only (secondary flows are ignroed).
pc	Pourpoint Catchments	The full drainage catchment for each pourpoint. These will overlap.
ccr	Pourpoint Catchment Containment Relaionthips	A matrix that represents the containment relationship between catchments (i,j). Valid values include: null there is no relationship between catchments 1 catchment i contains catchment j -1 catchment i is contained by catchment j
sc	Subcatchments	A single polygon is generated for each pourpoint created by assigning each elementary catchment to the most upstream pourpoint it is associated with, then merging the elementary catchments into a single polygon.
scr	Subcatchment Flow Relationships	The upstream/downstream relationship between the non-overlapping catchments (i,j). Valid values include: null there is no relationship between catchments 1 i is upstream of j -1 i is downstream of j
pc	Partitioned Catchments	Polygons generated by breaking the non-overlapping catchment polygons where they meet other pourpoint catchments.
pcr	Partitioned Catchment Flow Relationships	The upstream/downstream relationship between the transversal compliant catchments (i,j). Valid values include: null there is no relationship between catchments 1 i is upstream of j -1 i is downstream of j

prt	Pourpoint Relationship Tree	A tree representing the relationship between pourpoints traversing the primary flow network, starting at the most downstream nodes.
ic	Interior Catchments	A layer that includes all the holes (areas that are not connected to the flow network) within the pourpoint catchments. Holes are aggregated.

Outputs
Results are available in JSON or GeoPackage output.

GeoPackage output only includes spatial layers, not relationship matrix tables. Spatial layers include Output Pourpoints, Pourpoint Catchments, Non-Overlapping Catchments, Transversal Compliant Catchments and Interior Catchments.

JSON results include all requested layers in a single array of JSON Objects. At a minimum each JSON object has a key (the output type key) and a name (output type name). Spatial layers include feature collection details and are structured as GeoJSON. Relationship layers include the matrix headers as an array and the values as an array of arrays.

2.3 Parameters

In addition to the parameters identified in the above example services, some parameters would be generally available:

- srs
  - An OGC-style SRS specification such as “EPSG:4326”. Features returned will be reprojected into the requested SRS, and any spatial parameters (point or bbox) will be interpreted in the requested SRS.
- maxFeatures
  - Limits the number of returned features. Where possible and reasonable, the most relevant features are returned in order of decreasing relevance.
- bbox
  - Limits the features returned to only those which overlap with the given bbox
- idOnly
  - Limits the feature output to only a list of ID values for the requested features.

2.4 Content Types and Negotiation

The content type of the responses can be specified using either the “Accept” HTTP header, or by specifying the format extension in the URL. The URL format extension takes precedence, allowing for easy testing and interaction with the API using a standard web browser.

Supported content types:

Content type(s)	URL Format Extension(s)	Format Returned
text/html, application/xhtml+xml	.html, .xhtml	HTML
application/geo+json, application/json	.json, .geojson	GeoJSON
application/javascript	.jsonp, .geojsonp	GeoJSON wrapped in jsonp function call
application/geopackage+sqlite3	.geopackage	GeoPackage *Only spatial layers are included in geopackage output. Non-spatial layers (e.g. relationship matrices in Pourpoint service) are not included in GeoPackage output.*

3 Test Cases

A collection of test cases exist in the code repository to test the above API services. These test cases exists in the /src/test folder inside the repository. Details about the test cases can be found by viewing the code in the repository.

3.1 Test Data

All sample test data resides in the /src/test/resources folder. Currently the test cases make use of two test datasets:

- src/test/resources/data - the Richelieu area dataset used by the original application
- src/test/resources/data_small - a very simple dataset that mimics the Figure 3 provided in the NRCan-5000040751 contract.

Test case expected results are stored either with the Java test case, or in cases where the results were large, in json files in the src/test/results.

3.2 Test Case Overview

All test cases are defined in the BasicTestSuite java file. At the time of writing the following test classes exist. Additional details can be found in the code.

Test Case	Description

DrainageTest	Tests the upstream and downstream Catchment (Drainage Area) services for a few randomly picked points. Runs on the Richelieu test dataset.
ElementaryDrainageTest	Tests the upstream and downstream Elementary Catchment services for a few randomly selected points. Runs on the Richelieu test dataset.
FlowpathTest	Tests the upstream and downstream Elementary Flowpath services for a few randomly selected points. Runs on the Richelieu test dataset.
MultiDimensionalDownstreamTest	Tests the upstream and downstream Waterbody Flows services for a few randomly selected points. Runs on the Richelieu test dataset.
PourpointProjectionTest	Tests the projection of pourpoints to flowpath network based on the pourpoint ccode. A small collection of input points were selected attempting to represent some of the different data cases. Runs on the Richelieu test dataset.
PourpointSecondaryTest	Tests Pourpoint Service with specific focus on areas that contain secondary flows. Runs on the Richelieu test dataset.
PourpointTest	Tests Pourpoint Service and output layers. Runs on the Richelieu test dataset.
SimpleDataPourpointTest	A simple (and fast) test case that represents the example provided in the NRCan-5000040751 contract. This runs on the data_small test dataset.
StreamOrderTest	Tests the assignment of strahler, horton and hack order to stream network. This tests all flowpaths in the Richelieu test dataset.