Accessing EN Data via REST: Data Consumer Perspectives

Stuart Chalk, *University of North Florida*

2015 Exchange Network National Meeting
Supporting the Business of Environmental Protection

September 29–October 1, 2015
Sheraton Philadelphia Society Hill Hotel
Philadelphia, Pennsylvania

http://www.exchangenetwork.net/en2015

ABSTRACT

In designing REST interfaces, the EN should be cognizant of, and obtain feedback from, users outside of EN. Providing a robust, functional, and open REST based website and RESTful API to access the available data is key to its broad use.

Users, Users, Everywhere…

- Scientists are becoming more familiar with working with large datasets…

- …and are thus asking more interesting questions that can only be evaluated with such data

- And they want to be able to cite the dataset

My Perspective

- Environmental Analytical Chemist and Cheminformatician
- I need quality data to compare my results to
- I want to look at trends over time
- I want to map and overlay it with other GIS data
- I want to run statistics and find correlation
- I NEED DATA!

Specific Use Case

- Lower St. Johns River Report
- Analyze data about 29 tributaries of the Lower St. Johns River
- Report data to the public
- Provide interactive website to display data

http://www.sjrreport.org
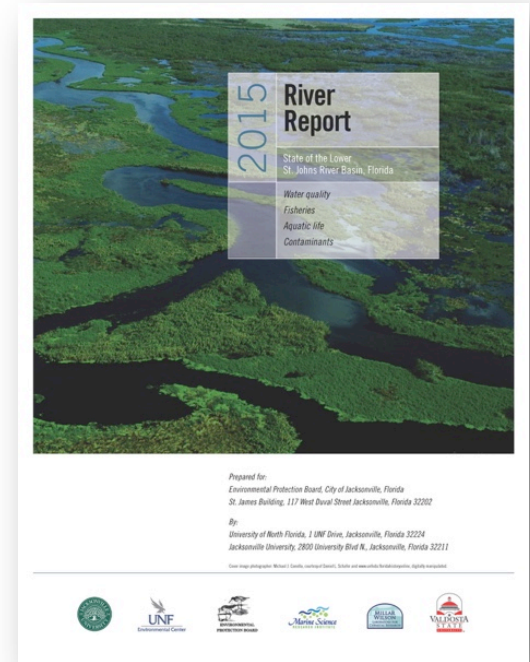
Available Water Data

- STORET
  http://ofmpub.epa.gov/storpubl/dw_pages.querycriteria

- WQX SOAP endpoints are no longer available so users are directed to…http://www.waterqualitydata.us/

What I Want When I Access EN WQX Data (or any EN data for that matter…)

REST Data Access By

- Organization
- Activity
- Location
- Characteristic
- Datatype (source)
- Medium (air, water)

- Datetime/Timestamp
- Dataset
- Program
- Other perspectives by which users would want to find data

Multiple Access Points

- /characteristic/station/datetime
- /station/characteristic/datetime
- /datetime/characteristic/station

- All do the same search under the hood and therefore retrieve the same data but organize it in different ways

Output Data Formats

- CSV

- Tab Delimited

- Excel

- KML

- XML*

- JSON**

- Choice of format available by content negotiation as well as option on URL
  - "application/json" in "Accept Header" with /char/arsenic/station/xyz1/results -OR-
  - /char/arsenic/station/xyz1/results?format=json

- Include compression in format choices – one less parameter in URL
  - results?format=zipcsv   (better)
  - results?format=csv&zip=yes

Standardized REST URL Meanings

- /org => info about organization & available data
- /org/activities => list of activities
- /org/stations => list of stations (other names?)
- /org/analyses & /station/analyses =>
  stats on data by characteristic
- /org/<oid>/station/<sid>/activities =>
  list of activities at station <sid> by org <oid>

Options for Output Data

- All of the following output list of characteristics, but the formats are different
- /chars => alphabetical list of character names
- /chars/detail => list with more detail
- /chars/byType => grouped by type
- /chars/counts => list with name and result count

Beware of URL Encoding

- Avoid characters that have to be encoded
  - ASCII Control Characters
  - Non-ASCII characters
  - Reserved characters ($ & +, / : ; = ? @)
  - Unsafe characters (space ' " < > # % { } | \ ^ ~ [ ] `)
- URLs may get broken and are difficult to deal with in scripting languages (encoded/unencoded)

Working Example

- Here is what it might look like…

Learn From Others

- PubChem (from NIH)
  - Power User Gateway (PUG) REST API
    https://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST.html
- If you talk to PubChem (Evan Bolton - bolton@ncbi.nlm.nih.gov is the lead) and I am sure get some valuable input

Comments

- One REST API for all of EN…
- …one for each logical group of flows (water)…
- …and for each separate flow
- Where the data comes from (servers) does not matter…
- Endpoints should display HTML that tells user what they can search for and how to write the URL
- Document the API clearly and completely with working examples
- Provide access to controlled vocabulary of every field that has one
- Provide unique identifiers that are persistent so that if the name of a characteristic is updated its data can still be found by its ID
- Provide snapshot timestamp so users can get new and changed data