

CS461 Fall 2017 Problem Statement

River Hendriksen

Abstract

Our projects purpose is to create a response handler for OPeNDAP (DAP 2) that allows integration with CoverageJSON. Our purpose in the project is to work with NASA Jet Propulsion Labs (JPL), OPeNDAP, and CoverageJSON to understand that requirements for porting NASA data, coded in Scientific formats, into data that can be presented to end users. These end users will be anyone interested in developing an application that can use this data. By the end of the Project there will be a full implementation of CoverJSON in OPeNDAP that has been tested and can be pushed to end users.

I. PROBLEM

As it stands NASA JPL, and OPeNDAP for that matter, do not have a system that reformats current scientific data into CoverageJSON. CoverageJSON is an extended library of JSON, JavaScript Object Notation, a standard format that is both human and machine readable that has become a crucial for the way data is transferred and received within the Internet, and web applications. To define Coverage, take the definition from the creators, feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain. CoverageJSON use is to allow data from such sources as satellites, weather, etc. to be entered in the JSON standard. Its main purpose is to, make it easier for application developers to consume coverage data in their applications.

II. PROPOSED SOLUTION

The proposed solution to this problem is to integrate CoverageJson into the OPeNDAP framework and work with NASA JPL to push their data out in the CoverageJson format. This will be done with a few steps:

- Working with NASA JPL and OPeNDAP to understand their needs and current systems.
- Designing a response handler in C++ to push NASA Scientific Data to end users in CoverageJSON format.
- Implementing tests and documentation for the created code and presenting it within Github.

The first steps begin with contact from NASA JPL, and OPeNDAP to understand the frameworks currently in place, along with discussing current proposed solutions by both teams to get a better understanding of the limits of both. NASA JPL currently uses OPeNDAP to present their data in other formats, but not in any as standard as CoverageJSON. After discussions the project will need documentation from all groups to get a well-rounded understanding of how all the systems work. This compromises most of the early work that needs to be done as being able to understand and present the current systems, along with how to efficiently use them, is of the utmost importance if this project is to be pushed to production.

Second, the design and creation of the handler in C++. This is likely to be the most difficult portion of the work as there is currently no code in place for such a handler and will require strong knowledge of all three systems (NASA, OPeNDAP, CoverageJSON) to be created. This is expected to take place early 2018. The handler must also be proven to work with all NASA data formats if it is to be pushed to a production environment.

Finally, testing and documentation must be created to assure the dependability of the code that was written, as well as allowing someone to take up the mantle once the project is finished (since this is an open source project). Testing/documentation will be a part of the final stage of development before a production push. As such this is arguably one of the most important steps as it is the last chance to find bugs and create a worthwhile handler for users. Documentation must be well written and followable by anyone in the open source community who would wish to change it or use it.

III. PERFORMANCE METRICS

Performance of the project can be decided in a manner of ways:

- A project website with readable documentation is presented to Github.
- There is a C++ solution that is used by OPenDAP to have a CoverageJSON response handler.
- Tests and documentation for C++ code that is presented to OPenDAP.
- A poster to the American Geophysical Society

All four conditions must be met for this project to be considered a success. As this is an open source project a website within GitHub is crucial as it allows anyone to be able to edit and reformat the code, to either improve or use as they deem. Without a GitHub page (or similar web page) this could not be considered an open source project. The C++ solution to OpenDAP is a must, as it in the main portion of this project, the success lingers on whether the solution can be implemented in OpenDAP and how well the implementation works. Without a solution, even the simplest of ones, the project will be considered a failure. Tests and documentation are also considered critical, if the project cannot be proved to work then it cannot be pushed to a production environment. If tests are not passed it is not as much of a problem as failing to provide a C++ solution, as tests can be administered on a later date. However, this is not to say that they should be pushed off. Lastly, a poster is to be present to the American Geophysical Society (AGS). This is not as important as the other parts of the project but should not be ignored. The point of the AGS poster is to show the community that this handler exists and can be widely used, think of it as marketing for the handler. Other performances can be attributed to the presentation in June at Oregon State University as to whether the project was successfully planned and presented.