Building rich and interactive web applications with CoverageJSON

Jon Blower, Maik Riechert, Guy Griffiths, Mridul Kumar, Riley Williams

Institute for Environmental Analytics, University of Reading, UK
Introduction

- Web browsers are becoming increasingly capable as visualisation and analysis platforms
- Lots of tools and libraries are built around images and “simple features”
  - GeoJSON, KML, OpenLayers, Leaflet ...
- Formats and tools for scientific / meteorological data are not always web-friendly
  - Complex, binary, desktop-oriented
  - Large variety, usually community-specific

=> Lots of people building ad-hoc solutions for web applications

- We want to bring scientific data within the reach of more Web and mobile app developers
  - Web-friendly formats (i.e. JSON)
  - More powerful and reusable visualisation/analysis tools
“Coverages”: a unifying concept

data = f(position, time, ...)

[Image of maps and graphs]
The CoverageJSON data format

- Rich and efficient JSON encoding of coverage data
- “As simple as possible but no simpler“
- Gridded and non-gridded data
- n-Dimensional data
- Continuous and categorical data
- Internationalisation
- Embedded semantics
  - (some interoperability with RDF through JSON-LD)
Skeleton CoverageJSON document

```json
{
    "domain": {
        ...
        "referencing": [ ... ]
    },

    "parameters": {
        "SST": { ... },
        "sea_ice": { ... }
    },

    "ranges": {
        "SST": { ... },
        "sea_ice": { ... }
    }
}
```

- Coordinates of data points and referencing information
- Metadata describing data values
- Data values as nD arrays
Metadata sample

```
{
    "observedProperty": {
        "id": "http://vocab.nerc.ac.uk/standard_name/sea_surface_temperature/",
        "label": {
            "en": "Sea Surface Temperature",
            "de": "Meeresoberflächentemperatur"
        },
        "description": { ... }
    },
    "unit": {
        "label": {
            "en": "Degree Celsius",
            "de": "Grad Celsius"
        },
        "symbol": {
            "value": "Cel",
            "type": "http://www.opengis.net/def/uom/UCUM/"
        }
    }
}
```
Scalability through tiling

• Large data files can be split into several JSON documents
• Each document holds part of the nD array
• Reduces need to run complex servers (cf. Web Map Tiling)
Interactive, in-browser reclassification of land cover maps
youtu.be/dxfmTkBdn90
In-browser intercomparison of models and observations
NASA Web World Wind and CovJSON

https://webworldwind.org
Beyond visualisation: Big Data analytics over the Web

- Put CovJSON tiles on a web server
  - or content delivery network
- Write simple analysis script in Python
  - Use Dask to treat tiles as single virtual dataset
  - Dask automatically downloads only the required tiles

=> work on big datasets more easily, without the need for a complex server

Calculate mean sea surface temperature over certain region:

```python
dataset = getDataset("http://my.covjson.doc")
sst = dataset["sst"]
result = da.mean(sst[0,:450,:]).compute()
```
CoverageJSON Resources

The CoverageJSON Format Specification

**Authors**
Marek Rucinski (University of Reading), Jan Bleck (University of Reading)

**Revision**
0.2 draft

**Date**
ww 29 2019

**Abstract**
CoverageJSON is a prospectus coverage data exchange format based on JavaScript Object Notation (JSON).

**Copyright**
© 2015 by ... This work is licensed under a Creative Commons Attribution 4.0 International License.

**TODO**
The following items are major outstanding issues to be resolved for the final version:

- #10: Representation of multiple time series.
- #7: Version number exclusivity-resolution of format.

1. **Introduction**

CoverageJSON is a format for encoding coverage data like grids, time series, and vertical profiles, distinguished by the

---

**Specification**

**Cookbook**
(start here!)

**Playground**

https://covjson.org
Tools
(https://covjson.org/tools)

• JavaScript data-reading library
• Visualisation
  – Leaflet plugin
  – Web World Wind demos
• Conversion
  – Python library to convert from NetCDF to CovJSON
  – Java libraries
• Servers
  – Export CoverageJSON from ncWMS/THREDDS
CovJSON vs OPeNDAP etc

• OPeNDAP can also deliver data to web clients
  – Binary format
  – Requires special server (e.g. Hyrax, THREDDS)

• CovJSON has pros and cons vs OPeNDAP
  – More friendly format for web developers
  – Can be served as static documents for scalability (with tiling)
  – Better support for semantic content
  – Less efficient (but compression helps a lot)

• Remember: CovJSON is just a format, which can be created in many different ways
  – On-the-fly or statically
  – Hence can be used as output format from THREDDS, WCS etc.
Conclusions

• CoverageJSON is a *simple but not simplified* format
  – Handles many kinds of data, include satellite images, derived products, in situ observations, numerical model data ...
  – A bit like a JSON version of NetCDF, with enhancements
  – Friendly for web developers
  – Supported by documentation, tools and examples

• Will be published soon as joint OGC/W3C document

• Potential future output format for Web Coverage Service

• We want to enable the community to generate new and exciting data-driven websites and apps!
Thank you!

@Jon_Blower
https://covjson.org
http://www.melodiesproject.eu