cdisc

2022
US
INTERCHANGE
26-27 OCTOBER | AUSTIN

COSA Dataset-JSON Hackathon Results

Sam Hume, DSc
VP, Data Science CDISC
Session 6, Tack B: Business Optimization & Technical Topics
Oct. 27, 2022

Meet the Speaker

Sam Hume

Title: VP, Data Science

Organization: CDISC

Sam Hume leads the CDISC Data Science team, which collaborates with CDISC staff and stakeholders to develop tools and standards that support clinical and translational data science. Sam directs delivery of the CDISC Library metadata repository that houses all CDISC standards, co-leads the CDISC Data Exchange Standards team, co-leads CORE, and leads the technical CDISC RWD efforts. He has 25 years' experience in clinical research informatics and has held a number of senior technology positions in the biopharmaceutical industry. He holds a doctorate in information systems.
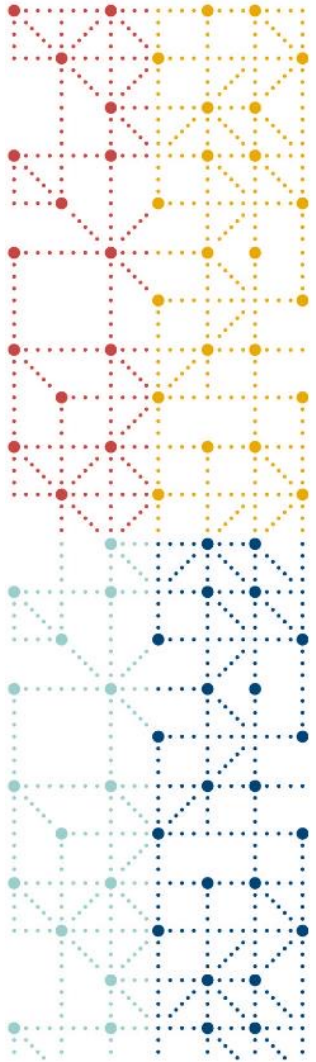
Agenda

1. COSA Dataset-JSON Hackathon Overview
2. Open-source Solutions
3. Commercial Solutions
4. Conclusion
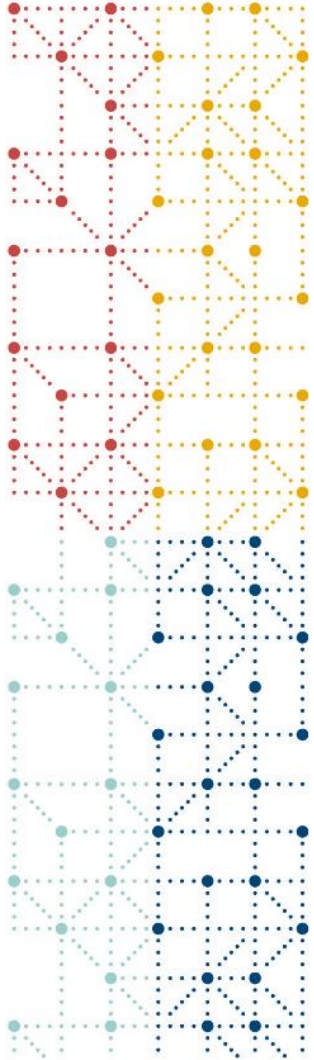
COSA Dataset-JSON Hackathon

An overview of the COSA Dataset-JSON Hackathon

CDISC Open-Source Alliance (COSA)

COSA Mission: The CDISC Open-Source Alliance (COSA) supports, promotes, and sometimes sponsors open-source and free software development projects that create tools for implementing or developing CDISC standards to drive innovation in the CDISC community.

- Virtual hackathon

- Dataset-JSON Hackathon solutions may apply to be included in the COSA Repository Directory

- Requires an open-source license

- Requires a public repository

- Conference session at the US Interchange will highlight solutions

- COSA Webinar to demo solutions

cdisc

https://cosa.cdisc.org/

Welcome to the COSA Dataset-JSON Hackathon

~ 150 registered participants

New draft data exchange standard for datasets

Create open-source solutions

Focused on data exchange

Convert Dataset-JSON to other common formats

Demonstrate and improve Dataset-JSON

cdisc

The Dataset-JSON Draft Data Exchange Standard

- Dataset-JSON is a draft standard for exchanging tabular datasets using JSON

- It is part of the ODM v2.0 draft standard
  - Planning to start Public Review on Nov. 8th

- It is based on the Dataset-XML v1.0 specification with enhancements, including
  - Much smaller file sizes
  - The addition of essential metadata to support data browsing

- Dataset-JSON links to a Define-XML file for the complete metadata

- Designed to meet the requirements of the regulatory submission use case
  - As well as other data exchange scenarios

cdisc

7

Virtual Hackathon: Key Dates

1 Sept

Hackathon Kickoff TC

17 Oct.

Share Project Results
Slides due Oct. 17th

26–27 Oct.

US Interchange
Presentation of Results Oct. 27th

Team Bi-Weekly Check-in: Wed 10am ET
• Sept. 7th
• Sept. 21st
• Oct. 5th
• Oct. 19th

7 Sept – 19 Oct.

Project public repo available with code for an initial release

Development may continue

Webinar Demonstrations

19 Oct.

8 Dec.

cdisc

Solutions Created during the Hackathon

- Demonstrate conversion to and from different, language specific dataset formats
- Dataset browsers / viewers
- Methods for handling large datasets
- RESTful Web Services

Language	# Solutions
R	5
SAS	4
Python	5
JavaScript	4
Java	1
Swift	1
XSLT	1
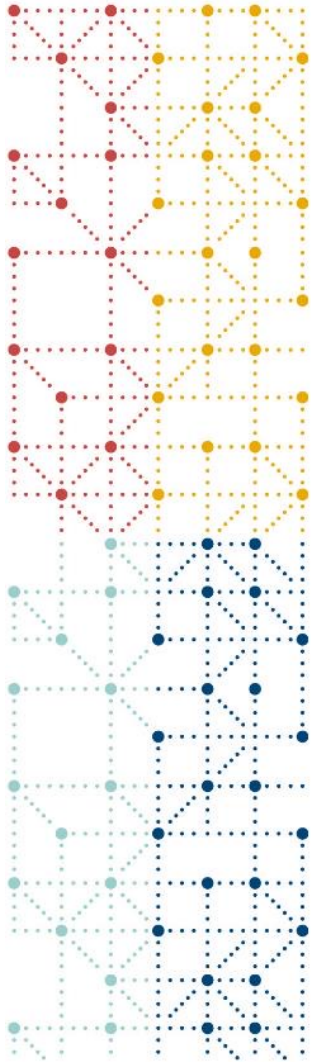
Open-Source Solutions

Summary of the Open-Source Solutions Developed During the Hackathon

Dataset-JSON – R package Implementation

- **Authors**: Mike Stackhouse (Atorus), Ben Straub(GSK), Eli Miller(Atorus), Eric Simms(GSK)

- **Repository**: https://github.com/atorus-research/dataset-json-hackathon

- **Website**: Read and Write JSON files specific to Clinical Trail Datasets • xportrjson (atorus-research.github.io)

- **Description**: Atorus and GSK built a simple R package {xportr} that writes out xpt files, we would like to extend this package to read and write out JSON files. This is a POC for that extension.

- **License**: MIT

R4DSJSON – R Package for Dataset-JSON

- **Author:** Ippei Akiya

- **Repository:**
https://github.com/i-akiya/R4DSJSON

- **Description:** R4DSJSON is to read CDISC Dataset-JSON files into R dataframe and to write it from R dataframe.

- **Purpose:** Make it easy to read and write Dataset-JSON in R.

- **License:** MIT

An example to read from dataset json and to display in data grid.

DM dataset example

```
dm <- R4DSJSON::read.dataset.json(
        file = system.file("testdata","dm.json", package="R4DSJSON")) %>%
        dplyr::select(-ITEMGROUPDATASEQ)

datatable(dm, options = list(pageLength = 5, scrollX='400px'))
```

Show 5 entries   Search:

	STUDYID	DOMAIN	USUBJID	SUBJID	RFSTDTC	RFENDTC	RFXSTDTC	RFXENDTC	
1	CDISCPILOT01	DM	CDISC001	1115	2012-11-30	2013-01-23	2012-11-30	2013-01-23	20
2	CDISCPILOT01	DM	CDISC002	1211	2012-11-15	2013-01-14	2012-11-15	2013-01-12	20
3	CDISCPILOT01	DM	CDISC003	1302	2013-08-29	2013-11-05	2013-08-29	2013-11-05	20
4	CDISCPILOT01	DM	CDISC004	1345	2013-10-08	2014-03-18	2013-10-08	2014-03-18	20
5	CDISCPILOT01	DM	CDISC005	1383	2013-02-04	2013-08-06	2013-02-04	2013-08-06	20

Showing 1 to 5 of 18 entries   Previous 1 2 3 4 Next

shineDSJSON

main analytics
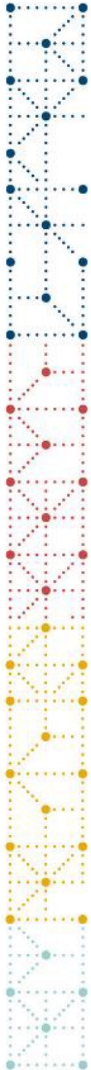
- lightweight **Dataset-JSON viewer** (app / browser)

- read in JSON files remote and local
- independent column and table wide regex search
- export table/search results as: CSV, Excel, Print

simple usage:

install_github("MichelLutz1994/shineDSJSON")

library(shineDSJSON)

shineDSJSON::runViewer()

Author: **Michel Lutz**
Feel free to checkout: https://github.com/MichelLutz1994/shineDSJSON

cdisc

CDISC Over Linked Data (COLD counterpart to FHIR)

1. Apply 1-line **JSON-LD** context URL to Dataset-JSON

```
{
    "@context": "https://mdr.cdisc.org/transfer_104ab4/define_BS1234_v2#",
    "clinicalData": {
```


Dataset-JSON expressed as RDF using JSON-LD

2. Point it to an explicit **JSON-LD** manifest/Define API
   - Common data contract
     - No need to include Define-XML files
     - Single *a-priori* source of metadata truth for all parties in a 'neutral zone'
     - Acts as standard machine-readable Data Transfer Agreement (DTA)
   - Transforms Dataset-JSON into a graph
     - Applies universally-unique ID to Dataset-JSON content
     - Express Dataset and Define as Linked Open Data on the semantic web
     - Express Dataset and Define as RDF triples / n-quads (to load into a common metamodel e.g. ODM + Biomedical Concepts)

3. Profit
   - Guarantee consistency when generating/interpreting datasets
   - Stream / preview large datasets without needing the entire JSON file
   - Explicit, native, human-readable format for expressing CDISC as linked data

Author: Jeremy Teoh   **Repo (exploratory)**: https://github.com/TeMeta/Dataset-JSON_hackathon

Smart Submission Dataset Viewer

The Open Source "Smart Submission Dataset Viewer",
a "smart" dedicated viewer for SDTM, SEND and ADaM datasets,
accepts submission datasets in Dataset-JSON, Dataset-XML and CSV format
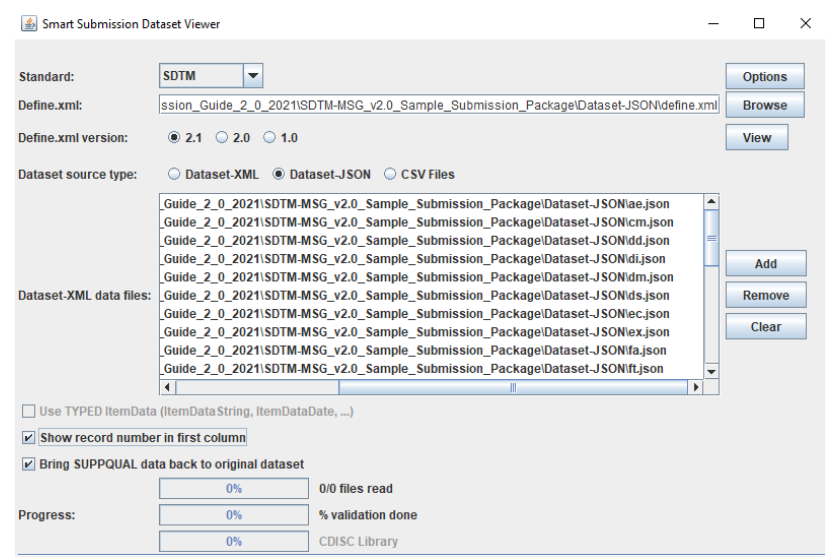
Included:
XPT datasets
can be
transformed to
Dataset-JSON
or Dataset-XML

Author: Jozef Aerts   https://sourceforge.net/projects/smart-submission-dataset-viewe/

RESTful Web Service using Dataset-JSON

- A simple prototype RESTful Web Service for querying submissions from a repository, using Dataset-JSON for the response, has been implemented

- Try it out at: http://xml4pharmaserver.com/WebServices/Submission_Services_Dataset-JSON/

Author: Jozef Aerts

Dataset-JSON - SAS Implementation

- **Author:** Lex Jansen

- **Repository:**
  https://github.com/lexjansen/dataset-json-sas

- **Description:** SAS macros and example
  programs used to create SAS datasets from
  Dataset-JSON as well as creating Dataset-
  JSON from SAS datasets.
  Metadata from Define-XML is read with Lua
  and used to for validation purposes and as
  pre-specified metadata.

- **Purpose:** Demonstrate Dataset-JSON's utility
  as a data exchange format.

- **License:** MIT

```sas
data _null_;
   length fref $8 jsonfile $200 code $200;
   did = filename(fref,"&root/json/&model");
   did = dopen(fref);
   do i = 1 to dnum(did);
      jsonfile = dread(did,i);
      if scan(lowcase(jsonfile), -1, ".") = 'json' then do;
         code=cats('%nrstr(%read_json(', jsonfile, ", &model);)");
         call execute(code);
      end;
   end;
   did = dclose(did);
   did = filename(fref);
run;


libname data "&root/data/&model";
ods output Members=members(keep=name);
   proc datasets library=data memtype=data;
quit;
run;

data _null_;
   length code $200;
   set members;
   name=lowcase(name);
   code=cats('%nrstr(%write_json(data.', name, ", &model);)");
   call execute(code);
run;
libname data clear;
```

SAS PROC XSL–Dataset-JSON from Define.xml

- **Author:** Pierre Dostie
- **Repository:**
  https://github.com/dostiep/Dataset-JSON
- **Description:** Create Dataset-JSON files using SAS Procedure XSL. SAS code is generated from a Define.xml using a XSL stylesheet (Dataset-JSON.xsl).
- **Purpose:** Demonstrate Dataset-JSON's utility as a data exchange format.
- **License:** MIT

```
filename xmlfile "<Your-path>\define.xml";
filename xslfile "<Your-path>\Dataset-JSON.xsl";
filename outfile temp;

proc xsl in=xmlfile xsl=xslfile out=outfile;
    parameter "libname" = "<Your-path>"
              "pretty" = "N";
run;

%inc outfile;

filename xmlfile clear;
filename xslfile clear;
filename outfile clear;
```

```
<xsl:template match="/">
    <xsl:variable name="studyOID" select="normalize-space($root/odm:Study/@OID)"/>
    <xsl:variable name="metaDataVersionOID" select="normalize-space($root/odm:Study/odm:MetaDataVersion/@OID)"/>
    <xsl:text>libname __tmp &quot;</xsl:text> <xsl:value-of select="$libname"/> <xsl:text>&quot;;</xsl:text>
    <xsl:value-of select="$lf"/>
    <xsl:value-of select="$lf"/>
    <xsl:text>%macro __checkds(__dsn);</xsl:text>
    <xsl:value-of select="$lf"/>
    <xsl:text>  %global __nobs;</xsl:text>
    <xsl:value-of select="$lf"/>
    <xsl:text>  %if %sysfunc(exist(__tmp.&amp;__dsn.)) %then %do;</xsl:text>
    <xsl:value-of select="$lf"/>
```

cdisc

Dataset-JSON React Demo

- **Author:** Katja Glaß

- **Repository:** https://github.com/KatjaGlassConsulting/dataset-json-react-demo

- **Description:** This is an example web application created by using the React framework to display the dataset content which is read in from an URL

- **Purpose:** demonstrate React usage and different table display options

- **License:** MIT

Dataset-JSON-viewer

Statskom CLINICAL PROGRAMMING

- **Author: Andrew Ndikom**

- **Repository:** GitHUB Repo

- **Description:** Built on the Datatables JS library the tool renders Dataset-JSON files in a tabular format and allows users to:
  - Filter rows,
  - Control pagination,
  - Show/ hide columns,
  - Export data to a number of common file formats,
  - Copy data.

- **Purpose:** Offer users an intuitive, modern, browser based, Excel™ like experience for viewing and interacting with Dataset-JSON files.

- **Licence: MIT**

Dataset-JSON viewer

Choose file | adsl.json
Dataset: Subject-Level Analysis, Data Type: clinicalData, Records: 254

Show 10 rows ▾ | Copy | Excel | CSV | PDF | Column visibility ▾

#	STUDYID (Study Identifier)	USUBJID (Unique Subject Identifier)	SUBJID (Subject Identifier for the Study)	SITEID (Study Site Identifier)	SITEGR1 (Pooled Site Group 1)	ARM (Description of Planned Arm)
1	CDISCPILOT01	01-701-1015	1015	701	701	Placebo
2	CDISCPILOT01	01-701-1023	1023	701	701	Placebo
3	CDISCPILOT01	01-701-1028	1028	701	701	Xanomeline High Dose
4	CDISCPILOT01	01-701-1033	1033	701	701	Xanomeline Low Dose
5	CDISCPILOT01	01-701-1034	1034	701	701	Xanomeline High Dose

swift-dataset-json: Swift Package for Dataset-JSON

- **Author:** Ippei Akiya

- **Repository:** https://github.com/i-akiya/swift-dataset-json

- **Description:** CDISC Dataset-JSON file reader in swift that is useful to develop a data review application on iPhone and iPad.

- **Purpose:** Make it easy to read Dataset-JSON in Swift.

- **License:** MIT

stream/serve/view-dataset-json

- **Authors:** Parexel (Juan Abdon, Ivan Osipov, Mauro Bringas, Dmitry Kolosov)

- **Repository:** stream/serve/view-dataset-json

- **Description:** This solution includes 3 subprojects
  - stream-dataset-json - Python library to read Dataset-JSON files as a stream
  - serve-dataset-json - Python library to serve Dataset-JSON files via API
  - view-dataset-json - TypeScript project implementing a viewer for Dataset-JSON files

- **Purpose:** The goal of the project is to write a library which allows to efficiently read Dataset-JSON files (including huge file sizes) and show to how it can be utilized for different purposes.

- **License: MIT**

Dataset-JSON - Python Implementation

- **Author:** Satish Ghadigaonkar

- **Repository:**
  https://github.com/satish-ghadigaonkar/pydsjson

- **Description:**
  - Python module to convert Dataset-JSON to Pandas dataframe, XPT and CSV as well as to convert XPT to Dataset-JSON.
  - Command-line interface is also available.

- **Purpose:** Demonstrate Dataset-JSON's utility as a data exchange format.

- **License:** MIT

```python
import pydsjson.dsjson

ds = pydsjson.dsjson.ReadDatasetJason(filepath=r".\examples\source\adlbc.json",
                                      item_group_prefix="")

# Covert to Pandas dataframe
df = ds.to_df(ds_name="ADLBC")

# Convert to XPT
ds.to_xpt(dest=r".\examples\output", ds_name="ADLBC",
          define=pydsjson.dsjson.ParseDefine(r".\examples\source\define.xml"))

# Convert to CSV
ds.to_csv(dest=r".\examples\output", ds_name="ADLBC")
```

cdisc

django-sdtm-export ([repository link](#))

Problem: When collecting clinical data in a Django app, implementing exports for each domain that adhere to SDTM standards can be laborious and repetitive.

Solution: django-sdtm-export: django-sdtm-export is an open source package that can be easily added to any django application. It allows for simple declarative export specifications, meaning new domains can be exported with as little as 75 lines of code. It supports exporting to both CSV and Dataset-JSON.

Contributors:

- [Lindus Health](#): Madeleine De Forest-Brown, Zaid Al-Jubouri, Amiel Kollek
- [Jeremy Teoh](#)

Technologies: Python, Django, Pandas

License: MIT

cdisc

Dataset-JSON Hackathon

- A open-source tool that search throughout the Dataset-JSON with user selections and convert it into various formats which make the conversion easier and more effective.

- **Description** : Created an open-source tool to work with Dataset-JSON, which takes JSON as input and allows the user to convert the desired output file format and vice versa. This tool is established to work with formats such as: SAS v5 XPORT, R data frame, xml, CSV.

- **Authors** : Renswick.D and Deepika.S from Pfizer (SPA) India

- **Software** : Python

- **Solution link**: renswick-pfizer/Dataset-JSON-Hackathon (github.com)

Hosted Python Notebook, JSONPath, CSV

- **Author**: Anthony Chow
- **Repo**: https://colab.research.google.com/drive/1myHpenokUEb4DXeghxbGTJr2rajwP0I8
- **Description**: A small data wrangling exercise using Google Colab, a hosted Python notebook.
- **Purpose**: Demonstration of downloading a Dataset-JSON file and verifying metadata using jasonpath-ng library. Investigate using open data visualization tool such as Google Public Data Explorer.
- **License**: MIT

Jupyter Notebook – Experimenting with Dataset-JSON

- **Author:** Sam Hume
- **Repository:**
  https://github.com/swhume/dataset-json-hackathon
- **Description:** Jupyter Notebook used to explore generating Python dataframes using Dataset-JSON as well as creating Dataset-JSON from CSV files. Also explores techniques for processing large datasets.
- **Purpose:** Demonstrate Dataset-JSON's utility as a data exchange format.
- **License:** MIT

Load Dataset-JSON using json module

For smaller datasets, simply load data using json module. This loads the entire file into memory

```
In [82]:   with open(data_file, 'r') as f:
               data = json.loads(f.read())
```

Show the name and label for the dataset as well as all the variables names that will be used as c
the data types in the Pandas dataframe.

```
In [83]:   dataset_attrs = list(data["clinicalData"]["itemGroupData"].values())[0]
           print(f"Name: {dataset_attrs['name']} ({dataset_attrs['label']})", end='\n\n')
           variables = [var['name'] for var in dataset_attrs['items']]
           print(f"Variables: {', '.join([var_name for var_name in variables])}")
           data_types = [var['type'] for var in dataset_attrs['items']]
```

```
Name: VS (Vital Signs)

Variables: ITEMGROUPDATASEQ, STUDYID, DOMAIN, USUBJID, VSSEQ, VSTESTCD, VSTEST, VSF
VSLOBXFL, VSREPNUM, VISITNUM, VISIT, EPOCH, VSDTC, VSDY
```

Create a dataframe from the Dataset-JSON file

Create a dataframe from the Dataset-JSON file. Then print the top 5 rows and provide the mem

```
In [84]:   df = pd.DataFrame(dataset_attrs['itemData'], columns=variables)
           print(df.head(5), end='\n\n')
           print(f"\ndataframe memory usage: {df.memory_usage().sum()} bytes")
```

```
   ITEMGROUPDATASEQ      STUDYID DOMAIN  USUBJID  VSSEQ VSTESTCD  \
0                 1  CDISCPILOT01     VS  CDISC001      1    DIABP
1                 2  CDISCPILOT01     VS  CDISC001      2    DIABP
2                 3  CDISCPILOT01     VS  CDISC001      3    DIABP
3                 4  CDISCPILOT01     VS  CDISC001      4    DIABP
4                 5  CDISCPILOT01     VS  CDISC001      5    DIABP
```
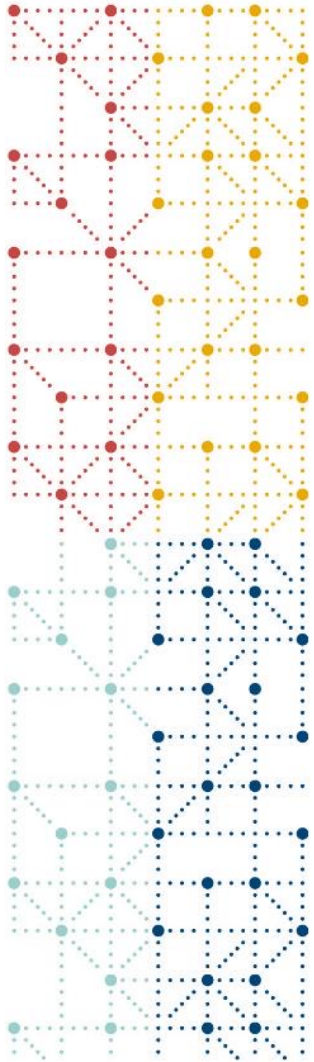
Commercial Solutions

Summary of the Commercial Solutions Developed During the Hackathon

SDTM-ETL

The commercial SDTM-ETL mapping software (for generation of SDTM and SEND datasets including the corresponding define.xml) allows to generate the submission datasets in XPT, Dataset-JSON, Dataset-XML and CSV format.
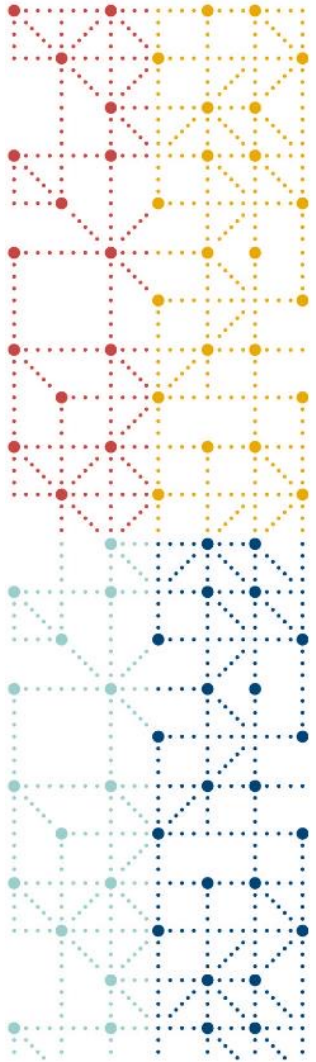

Author: Jozef Aerts

Website:

www.xml4pharma.com/SDTM-ETL

Conclusion

Summary of Hackathon and Future Events

Overall Impressions of Dataset-JSON

- Dataset-JSON works:
  - As a general data exchange
  - As a general dataset format
  - For use with web-based APIs
  - Works with a wide-range of programming languages and technology stack
  - Simple to process
  - Easy to transform into SAS datasets, R or Python dataframes, and CSV

- Dataset-JSON file sizes are smaller than SAS XPORT v5 and Dataset-XML

- Dataset-JSON is row-based - typically transformed into datasets for analysis

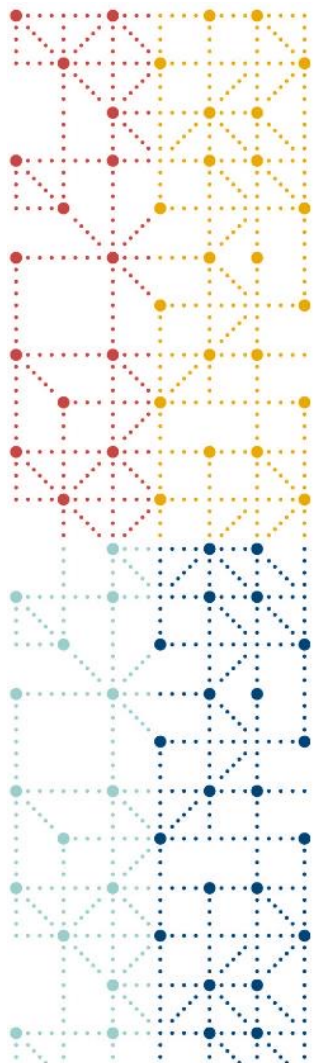- JSON is a language, platform independent data exchange format

Updates to Dataset-JSON

- Add useful ODM attributes:
  - ODMVersion, FileOID, PriorFileOID, CreationDateTime, AsOfDateTime…

- API reference implementation:
  - A common API achieves this better than individually-stored copies of a .XML document
  - Currently planned as a supplement to ODM v2.0

Next Steps

- Dec 8th COSA Spotlight Webinar demonstrating the Dataset-JSON Hackathon solutions

- Add Dataset-JSON Hackathon solutions to the COSA Repository Directory

- ODM v2.0 Public Review begins Nov. 8th
  - Dataset-JSON is part of ODM v2.0
  - Chance to comment

- Next COSA Hackathon: **admiral**
  - An open source, modularized toolbox that enables the collaborative development of ADaM datasets in R
  - Training: Tuesday, 17 January, 10AM - 1PM ET
  - Hackathon Kickoff: Thursday, 26 January, 10AM - 12PM ET
  - Hackathon: 1-28 February

Thank You!