Essential JSON Skills for Clinical Trial Programmers

Elliot Inman, PhD
Sr. Manager, SAS Health and Life Sciences R&D
PharmaSUG October 2, 2025

Overview

An Introduction to JSON

- Technical Definition of JSON
- Why clinical trial programmers should care
- Using JSON data
- Implementation in SAS9 and SAS Viya
- Sharing data with R and Python
- Viewers and Linters
- Fast Forward

JSON: Technical Overview

A Very Brief History

- “JavaScript Object Notation”
- Invented in the early 2000s by Douglas Crockford
- Modified MIT license in 2002
- Public Domain in 2022

"There's a lot of argument about how you pronounce that, but I strictly don't care.“

~ Crockford 2011

As cited at Wikipedia,
https://en.wikipedia.org/wiki/Douglas_Crockford
https://en.wikipedia.org/wiki/JSON

The Official JSON Standard

"**JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate."

https://www.json.org/json-en.html

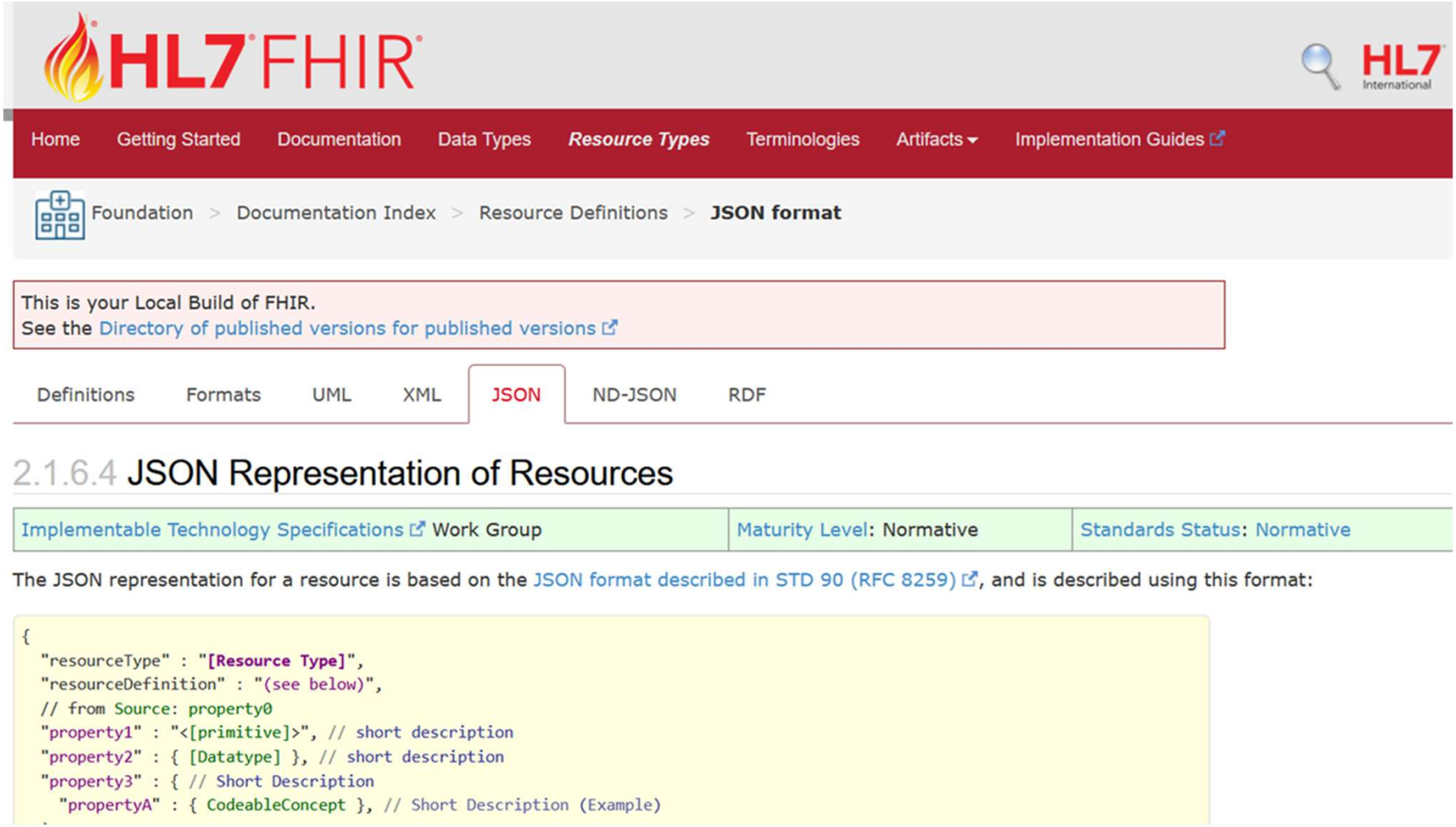
Example

{"SASJSONExport":"1.0","SASTableData+HEART":[{"Status":"Dead","DeathCause":"Other","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":29,"Height":62.5,"Weight":140,"Diastolic":78,"Systolic":124,"MRW":121,"Smoking":0,"AgeAtDeath":55,"Cholesterol":null,"Chol_Status":"","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Dead","DeathCause":"Cancer","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":41,"Height":59.75,"Weight":194,"Diastolic":92,"Systolic":144,"MRW":183,"Smoking":0,"AgeAtDeath":57,"Cholesterol":181,"Chol_Status":"Desirable","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":57,"Height":62.25,"Weight":132,"Diastolic":90,"Systolic":170,"MRW":114,"Smoking":10,"AgeAtDeath":null,"Cholesterol":250,"Chol_Status":"High","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Moderate (6-15)"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":39,"Height":65.75,"Weight":158,"Diastolic":80,"Systolic":128,"MRW":123,"Smoking":0,"AgeAtDeath":null,"Cholesterol":242,"Chol_Status":"High","BP_Status":"Normal","Weight_Stat

Why you should care

Standard for data exchange among a wide variety of pharma and healthcare data systems

- Real World Evidence
- Study Submission Data
- ClinicalTrials.gov



About Standards Tools Partnerships Education Events Membership

Home / Standards / Data Exchange / Dataset-JSON

Dataset-JSON

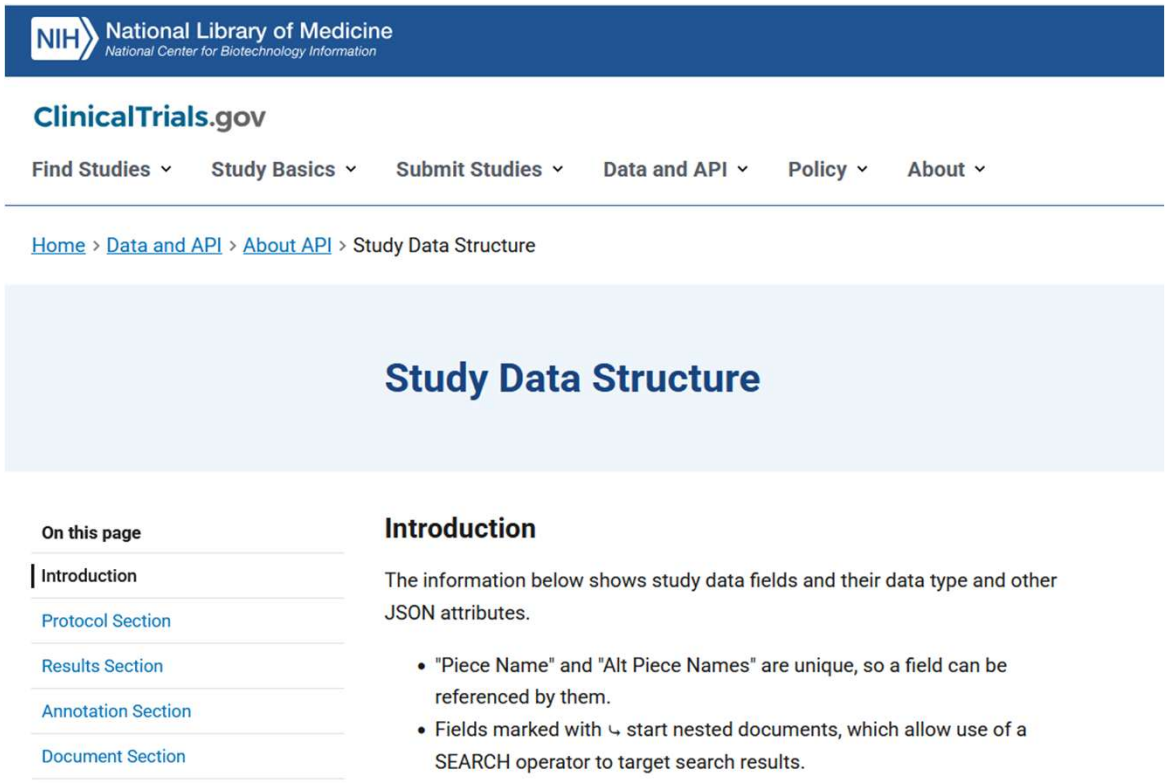
Description Versions Archive

Dataset-JSON is a data exchange standard for sharing tabular data using JSON. It is designed to meet a wide range of data exchange scenarios, including regulatory submissions and API-based data exchange.

Features of Dataset-JSON v1.1

https://www.cdisc.org/standards/data-exchange/dataset-json

ClinicalTrials.gov



https://clinicaltrials.gov/data-api/about-api/study-data-structure

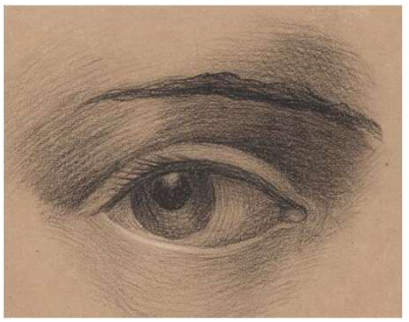
Using JSON Data

- Readability
- Reading and Writing
- Some quirks in the specification
- One Standard is never enough

Human Readable?


```
{"SASJSONExport":"1.0","SASTableData+HEART":
[{"Status":"Dead","DeathCause":"Other","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":29,"Height":62.5,"Weight":140,"Diastolic":78,"Systolic":124,"MR
W":121,"Smoking":0,"AgeAtDeath":55,"Cholesterol":null,"Chol_Status":"","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-
smoker"},
{"Status":"Dead","DeathCause":"Cancer","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":41,"Height":59.75,"Weight":194,"Diastolic":92,"Systolic":144,"M
RW":183,"Smoking":0,"AgeAtDeath":57,"Cholesterol":181,"Chol_Status":"Desirable","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"No
n-smoker"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":57,"Height":62.25,"Weight":132,"Diastolic":90,"Systolic":170,"MRW":1
14,"Smoking":10,"AgeAtDeath":null,"Cholesterol":250,"Chol_Status":"High","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Moderate
(6-15)"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":39,"Height":65.75,"Weight":158,"Diastolic":80,"Systolic":128,"MRW":1
23,"Smoking":0,"AgeAtDeath":null,"Cholesterol":242,"Chol_Status":"High","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-
smoker"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Male","AgeAtStart":42,"Height":66,"Weight":156,"Diastolic":76,"Systolic":110,"MRW":116,"S
moking":20,"AgeAtDeath":null,"Cholesterol":281,"Chol_Status":"High","BP_Status":"Optimal","Weight_Status":"Overweight","Smoking_Status":"Heavy
(16-25)"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":58,"Height":61.75,"Weight":131,"Diastolic":92,"Systolic":176,"MRW":1
17,"Smoking":0,"AgeAtDeath":null,"Cholesterol":196,"Chol_Status":"Desirable","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Non-
smoker"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":36,"Height":64.75,"Weight":136,"Diastolic":80,"Systolic":112,"MRW":1
10,"Smoking":15,"AgeAtDeath":null,"Cholesterol":196,"Chol_Status":"Desirable","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Mo
derate (6-15)"},
{"Status":"Dead","DeathCause":"Other","AgeCHDdiag":null,"Sex":"Male","AgeAtStart":53,"Height":65.5,"Weight":130,"Diastolic":80,"Systolic":114,"MRW":
99,"Smoking":0,"AgeAtDeath":77,"Cholesterol":276,"Chol_Status":"High","BP_Status":"Normal","Weight_Status":"Normal","Smoking_Status":"Non-smoker"},
{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Male","AgeAtStart":35,"Height":71,"Weight":194,"Diastolic":68,"Systolic":132,"MRW":124,"S
moking":0,"AgeAtDeath":null,"Cholesterol":211,"Chol_Status":"Borderline","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-
```

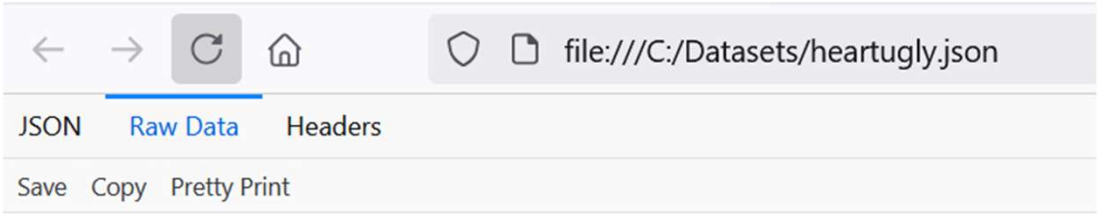
PROC EYEBALL;



VIEWTABLE: Work.Heart

	Status	Cause of Death	Age CHD Diagnosed	Sex	Age at Start	Height	Weight	Diastolic	Systolic
1	Dead	Other	.	Female	29	62.5	140	78	124
2	Dead	Cancer	.	Female	41	59.75	194	92	144
3	Alive		.	Female	57	62.25	132	90	170
4	Alive		.	Female	39	65.75	158	80	128
5	Alive		.	Male	42	66	156	76	110
6	Alive		.	Female	58	61.75	131	92	176
7	Alive		.	Female	36	64.75	136	80	112
8	Dead	Other	.	Male	53	65.5	130	80	114
9	Alive		.	Male	35	71	194	68	132
10	Dead	Cerebral Vascular Disease	.	Male	52	62.5	129	78	124
11	Alive		.	Male	39	66.25	179	76	128
12	Alive		57	Male	33	64.25	151	68	108
13	Alive		55	Male	33	70	174	90	142
14	Alive		79	Male	57	67.25	165	76	128
15	Alive		66	Male	44	69	155	90	130
16	Alive		.	Female	37	64.5	134	76	120
17	Alive		.	Male	40	66.25	151	72	132
18	Dead	Cancer	56	Male	56	67.25	122	72	120
19	Alive		.	Female	42	67.75	162	96	138
20	Dead	Coronary Heart Disease	74	Male	46	66.5	157	84	142
21	Alive		.	Female	37	66.25	148	78	110
22	Alive		.	Female	45	64	147	74	120
23	Alive		.	Female	59	65.75	156	74	156

https://www.loc.gov/pictures/resource/ppmsca.85136/




```
{
  "SASJSONExport": "1.0",
  "SASTableData+HEART": [
    {
      "Status": "Dead",
      "DeathCause": "Other",
      "AgeCHDdiag": null,
      "Sex": "Female",
      "AgeAtStart": 29,
      "Height": 62.5,
      "Weight": 140,
      "Diastolic": 78,
      "Systolic": 124,
      "MRW": 121,
      "Smoking": 0,
      "AgeAtDeath": 55,
      "Cholesterol": null,
      "Chol_Status": "",
      "BP_Status": "Normal",
      "Weight_Status": "Overweight",
      "Smoking_Status": "Non-smoker"
    },
    {
      "Status": "Dead",
      "DeathCause": "Cancer",
      "AgeCHDdiag": null,
      "Sex": "Female",
      "AgeAtStart": 41,
```

Implementation in SAS9 and SAS Viya

Two Methods

Libname for the JSON Engine

PROC JSON

LIBNAME Statement: JSON Engine

Provides read-only sequential access to JSON data.

Valid in:	Anywhere
Category:	Data Access
Notes:	The JSON file is read only once, when the JSON engine LIBNAME statement is assigned. To read the JSON file again, you must reassign the JSON libref.

Example in SAS9

```
filename mydata "C:\Datasets\heartpretty.json";
libname myjson JSON fileref=mydata;

proc datasets;
run;

proc print data = myjson.sastabledata_heart;
run;
```

The SAS System

Directory	
Libref	MYJSON
Engine	JSON
Access	READONLY
Physical Name	C:\Datasets\heartpretty.json

#	Name	Member Type
1	ALLDATA	DATA
2	ROOT	DATA
3	SASTABLEDATA_HEART	DATA

Obs	ordinal_root	ordinal_SASTableData_HEART	Status	DeathCause	AgeCHDdiag	Sex	AgeAtSta
1	1	1	Dead	Other	.	Female	2
2	1	2	Dead	Cancer	.	Female	4
3	1	3	Alive		.	Female	5
4	1	4	Alive		.	Female	3
5	1	5	Alive		.	Male	4
6	1	6	Alive		.	Female	5
7	1	7	Alive		.	Female	3
8	1	8	Dead	Other	.	Male	5
9	1	9	Alive		.	Male	3
10	1	10	Dead	Cerebral	.	Male	5

Convert JSON to SAS

```
36   data heart;
37       set myjson.sastabledata_heart;
38       run;

NOTE: There were 5209 observations read from the data set MYJSON.SASTABLEDATA_HEART.
NOTE: The data set WORK.HEART has 5209 observations and 19 variables.
NOTE: DATA statement used (Total process time):
      real time           0.04 seconds
      cpu time            0.00 seconds
```

VIEWTABLE: Work.Heart

	Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight
1	Dead	Other	.	Female	29	62.5	140
2	Dead	Cancer	.	Female	41	59.75	194
3	Alive		.	Female	57	62.25	132
4	Alive		.	Female	39	65.75	158
5	Alive		.	Male	42	66	156
6	Alive		.	Female	58	61.75	131
7	Alive		.	Female	36	64.75	136
8	Dead	Other	.	Male	53	65.5	130
9	Alive		.	Male	35	71	194

JSON Procedure

Syntax ▾ Overview Concepts Usage ▾ Examples ▾

Syntax

Table of Procedure Tasks and Examples

Syntax

PROC JSON OUT=*fileref* | *"external-file"* <*options* >;

EXPORT <*libref.*> *SAS-data-set* < (*SAS-data-set-options*) > < */options* > ;

WRITE VALUES *value(s)* < */options* > ;

WRITE OPEN *type* ;

WRITE CLOSE ;

Example in SAS9

```
data heart;
    set sashelp.heart;
    run;

proc json out="C:\heartugly.json";
    export heart;
run;

proc json out="C:\heartpretty.json" pretty;
    export heart;
run;
```

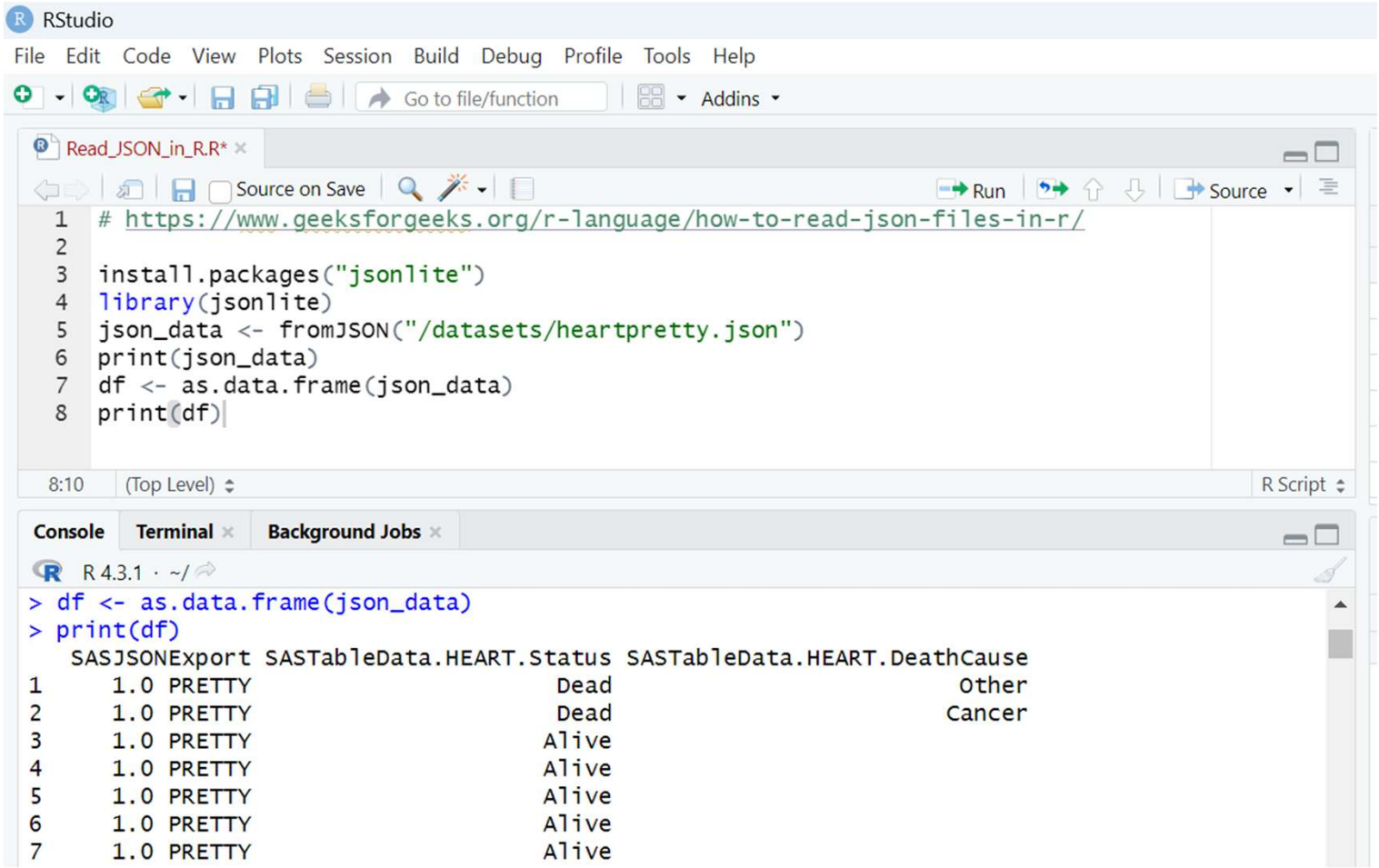
UGLY

{"SASJSONExport":"1.0","SASTableData+HEART":[{"Status":"Dead","DeathCause":"Other","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":29,"Height":62.5,"Weight":140,"Diastolic":78,"Systolic":124,"MRW":121,"Smoking":0,"AgeAtDeath":55,"Cholesterol":null,"Chol_Status":"","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Dead","DeathCause":"Cancer","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":41,"Height":59.75,"Weight":194,"Diastolic":92,"Systolic":144,"MRW":183,"Smoking":0,"AgeAtDeath":57,"Cholesterol":181,"Chol_Status":"Desirable","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":57,"Height":62.25,"Weight":132,"Diastolic":90,"Systolic":170,"MRW":114,"Smoking":10,"AgeAtDeath":null,"Cholesterol":250,"Chol_Status":"High","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Moderate (6-15)"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":39,"Height":65.75,"Weight":158,"Diastolic":80,"Systolic":128,"MRW":123,"Smoking":0,"AgeAtDeath":null,"Cholesterol":242,"Chol_Status":"High","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Male","AgeAtStart":42,"Height":66,"Weight":156,"Diastolic":76,"Systolic":110,"MRW":116,"Smoking":20,"AgeAtDeath":null,"Cholesterol":281,"Chol_Status":"High","BP_Status":"Optimal","Weight_Status":"Overweight","Smoking_Status":"Heavy (16-25)"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":58,"Height":61.75,"Weight":131,"Diastolic":92,"Systolic":176,"MRW":117,"Smoking":0,"AgeAtDeath":null,"Cholesterol":196,"Chol_Status":"Desirable","BP_Status":"High","Weight_Status":"Overweight","Smoking_Status":"Non-smoker"},{"Status":"Alive","DeathCause":"","AgeCHDdiag":null,"Sex":"Female","AgeAtStart":36,"Height":64.75,"Weight":136,"Diastolic":80,"Systolic":112,"MRW":110,"Smoking":15,"AgeAtDeath":null,"Cholesterol":196,"Chol_Status":"Desirable","BP_Status":"Normal","Weight_Status":"Overweight","Smoking_Status":"Moderate (6-15)"},{"Status":"Dead","DeathCause":"Other","AgeCHDdiag":null,"Sex":"Male","AgeAtStart":53,"Height":65.5,"

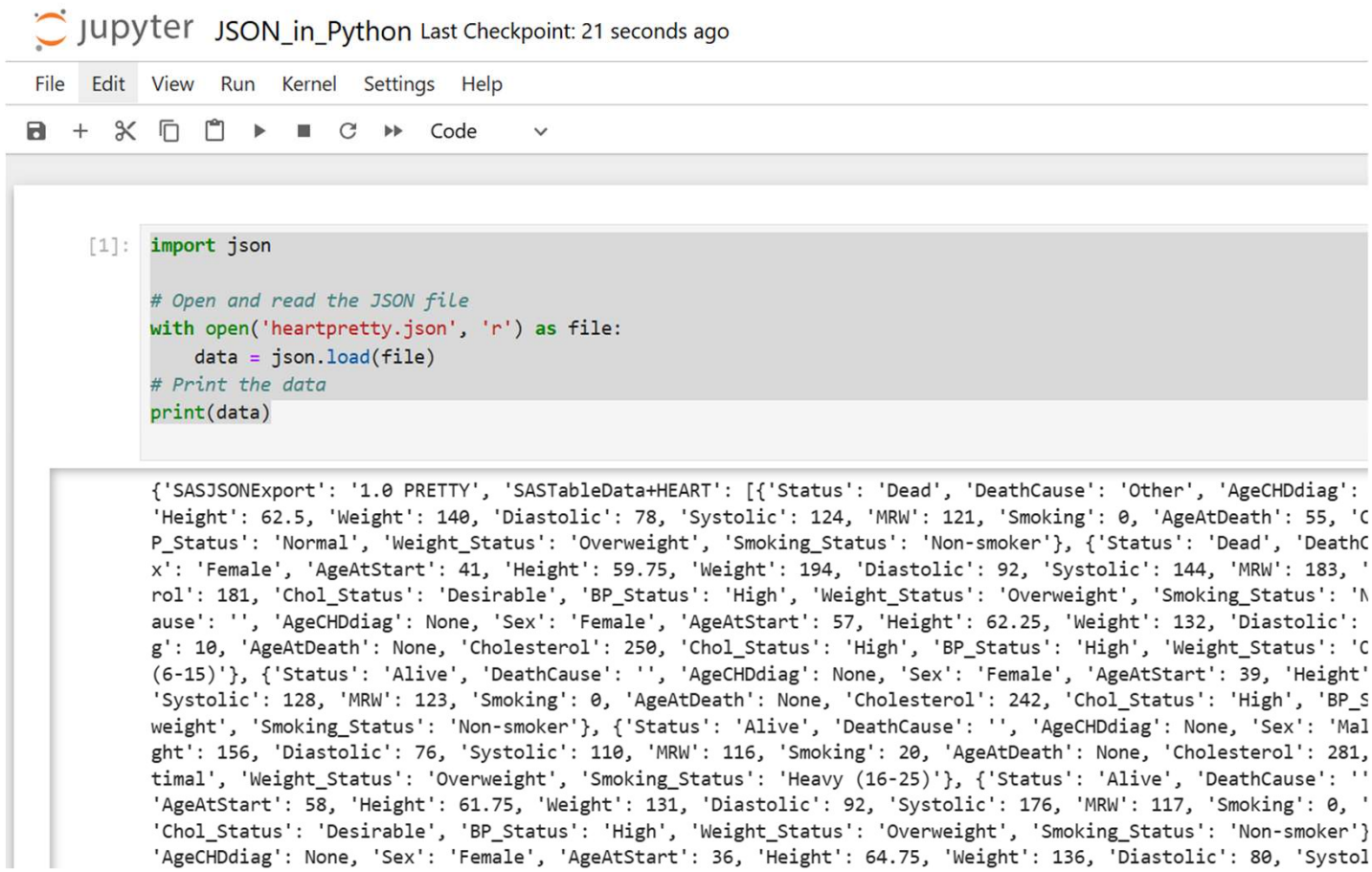
Pretty


```
{
 "SASJSONExport": "1.0 PRETTY",
 "SASTableData+HEART": [
  {
   "Status": "Dead",
   "DeathCause": "Other",
   "AgeCHDdiag": null,
   "Sex": "Female",
   "AgeAtStart": 29,
   "Height": 62.5,
   "Weight": 140,
   "Diastolic": 78,
   "Systolic": 124,
   "MRW": 121,
   "Smoking": 0,
   "AgeAtDeath": 55,
   "Cholesterol": null,
   "Chol_Status": "",
   "BP_Status": "Normal",
   "Weight_Status": "Overweight",
   "Smoking_Status": "Non-smoker"
  },
  {
   "Status": "Dead",
   "DeathCause": "Cancer"
```

Plays Nice with Others: R



Plays Nice with Others: Python



Some Unique Features of JSON

No Padding

No Defined Numeric Types

> "JSON is agnostic about the semantics of numbers. In any programming language, there can be a variety of number types of various capacities and complements, fixed or floating, binary or decimal. That can make interchange between different programming languages difficult. JSON instead offers only the representation of numbers that humans use: a sequence of digits."

Alternate Formatting

NDJSON

https://ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf

Viewers

Making Data Readable

- Proc print;
- Notepad
- Any html5 web browser
- MS Visual Studio Code

Lint (software)

18 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Lint is the computer science term for a static code analysis tool used to flag programming errors, bugs, stylistic errors and suspicious constructs.[4] The term originates from a Unix utility that examined C language source code.[1] A program which performs this function is also known as a "**linter**" or "**linting tool**".

Lint	
Original author(s)	Stephen C. Johnson
Developer(s)	AT&T Bell Laboratories
Initial release	July 26, 1978; 47 years ago[1]
Written in	C
Operating system	Cross-platform
Available in	English
Type	Static program analysis tools
License	Originally proprietary commercial software, now free software under a BSD-like license[2][3]

History [edit]

Stephen C. Johnson, a computer scientist at Bell Labs, came up with the term "lint" in 1978 while debugging the yacc grammar he was writing for C and dealing with portability issues stemming from porting Unix to a 32-bit machine.[5][1] The term was borrowed from the word lint, the tiny bits of fiber and fluff shed by clothing, as the command he wrote would act like a lint trap in a clothes dryer, capturing waste fibers while leaving whole fabrics intact. In 1979, lint programming was used outside of Bell Labs for the first time, in the seventh version (V7) of Unix.

https://en.wikipedia.org/wiki/Lint_(software)

Fast Forward: Dataset-JSON

CDISC Dataset-JSON repository

This repository contains the schema, specification, and examples of the Dataset-JSON standard for Clinical Datasets.

The Dataset-JSON v1.1 release can be downloaded as a zip file.

Dataset-JSON Specification

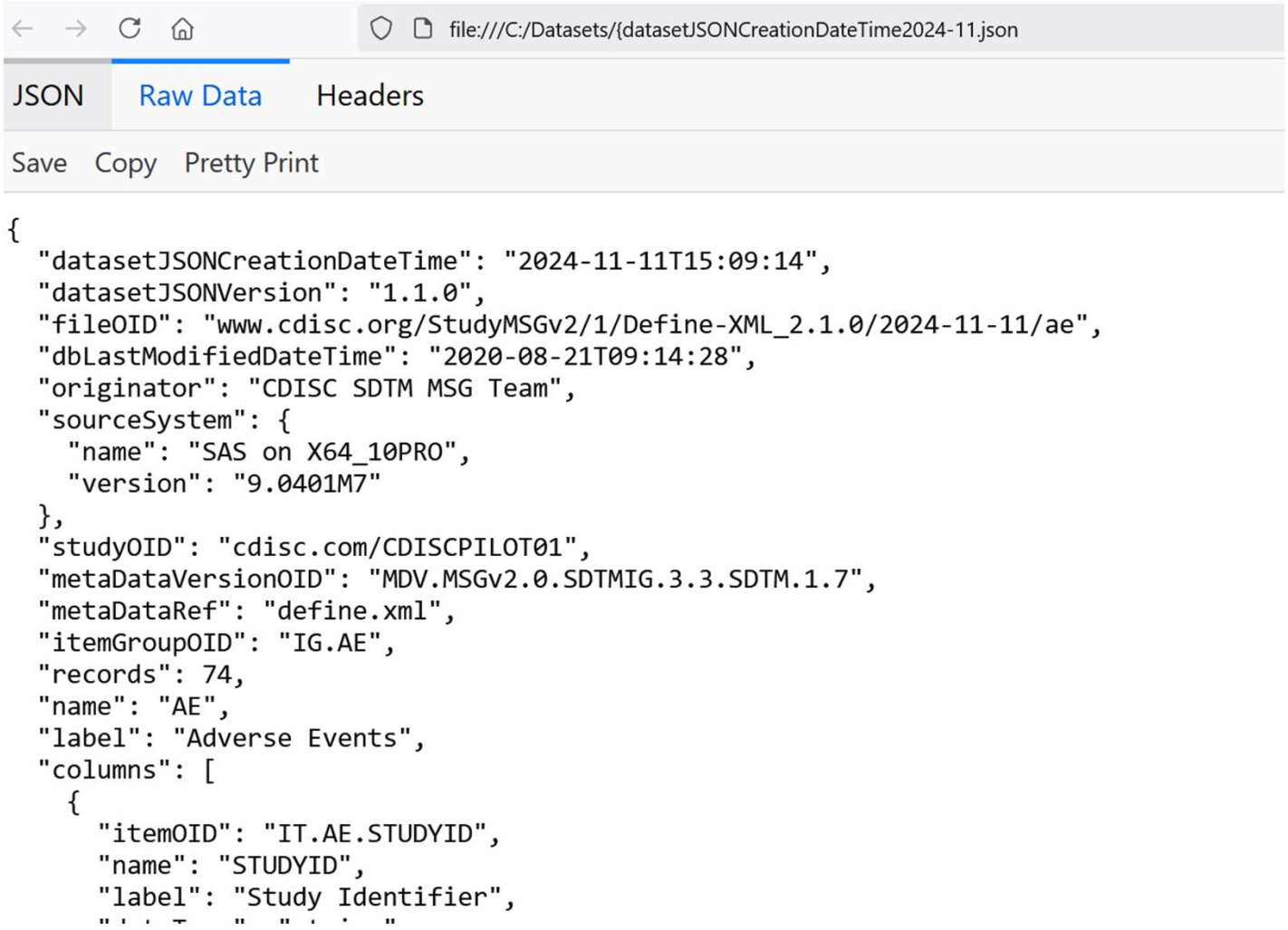
The doc folder contains the specifications of:

- Dataset-JSON v1.1.
- The NDJSON representation of Dataset-JSON v1.1
- Compressed Dataset-JSON v1.1

Dataset-JSON Schema

The schema folder contains the LinkML and JSON schemas for the JSON and NDJSON specification of Dataset-JSON.

https://github.com/cdisc-org/DataExchange-DatasetJson



March 2025 FDA Guidance

3.1.1 v5 Transport Format

The Transport Format (XPORT) Version 5 is the file format for the submission of all electronic datasets.[22] XPORT is an open file format published by SAS Institute for the exchange of study data. Data can be translated to and from XPORT to other commonly used formats without the use of programs from any specific vendor. There should be one dataset per transport file, and the dataset in the transport file should be named the same as the transport file (e.g., 'ae' and ae.xpt, 'suppae' and suppae.xpt, 'lb1' and lb1.xpt).

XPORT files can be created by the COPY Procedure in SAS Version 5 and higher of the SAS Software. SAS Transport files processed by the SAS CPORT cannot be reviewed, processed, or archived by FDA. Sponsors can find the record layout for SAS XPORT transport files through SAS technical document TS-140.[23] All SAS XPORT transport files should use .xpt as the file extension, and the files should not be compressed. Note also that SAS custom formats should NOT be used in submissions to the FDA.

Being able to read and write JSON for both humans and machine systems is an essential skill for any clinical trial programmer.

SAS makes it exceptionally easy to do so.

Thank You!

Elliot Inman

Elliot.Inman@sas.com

https://www.linkedin.com/in/elliot-inman