# The GPJSON Format Specification

Json Format for Geophysics Specification

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Authors | Email |
| 14/02/2015 | 0.1 | Aben Lee | aben.lee@foxmail.com |
| 05/05/2015 | 0.2 | Aben Lee | aben.lee@foxmail.com |
| 21/07/2017 | 1.0 | Aben Lee | aben.lee@foxmail.com |

|  |
| --- |
| **Abstract :**  GPJSON is a Geophysics Methods interchange format based on JavaScript Object Notation (JSON) , including Geophysical data acquisition, Data Processing etc.. It defines several types of json objects and manner smilar to NETCDF format, in which they are combine to represent data about geosciences features, their properties (attributes) and variable. Since it can exchange data with NETCDF format,It is also suited to other geosciences and research fields. |

**Note:** This document is a product of the Eageo Information & Exploration Technology Research [Institution](http://dict.youdao.com/w/research%20institution/#keyfrom=E2Ctranslation) (EIET) from Henan Polytechnic University (hpu), P.R.China.

Copyright Notice: Copyright (c) 2017 EIET and the persons identified as the document authors. All rights reserved.

1.Introduction

GPJSON is a json encode format in order to represents all kinds of geophysical data structure. A GPJSON object may represent date/time, Location, and the other informations. since it is developed on the NETCDF format, it also may represent the variables, dataset and their attributes. Features in GPJSON contain info object and collector object to entirely represent a dataset object. The MainInfo and MainCollectors is a options features for describing global informations a list of dataset object.

The format is concerned with geophysical data in the broadest sense; anything with qualities that are bounded in geophysical field might be a Feature whether or not it is a physical structure. The concepts in GPJSON are not new; they are derived from preexisting network Common Data Form (NETCDF) standards and have been streamlined to better suit web application development using JSON.

A complete GPJSON data structure is always an object (in JSON terms). In EgJSON, an object consists of a collection of name/value pairs -- also called members. For each member, the name is always a string. Member values are either a string, number, object, array or one of the literals: true, false, and null. An array consists of elements where each element is a value as described above.

2. Conventions Used in This Document

The ordering of the members of any JSON object defined in this document MUST be considered irrelevant, as specified by [RFC7159].

Some examples use the combination of a JavaScript single-line comment (//) followed by an ellipsis (...) as placeholder notation for content deemed irrelevant by the authors. These placeholders must of course be deleted or otherwise replaced, before attempting to validate the corresponding JSON code example.

Whitespace is used in the examples inside this document to help illustrate the data structures, but it is not required. Unquotedwhitespace is not significant in JSON.

GPJSON objects MUST NOT contain the defining members of other types.

Extensions MAY be used, but MUST NOT change the semantics of GPJSON members and types.

The media type for GPJSON is "application/gp+json"

3. Definitions

JavaScript Object Notation (JSON), and the terms object, member, name, value, array, number, true, false, and null, are to be interpreted as defined in [RFC7159].

Inside this document, the term of dataset type refers to a pair of InfoTree object and DataTable object, and the keys of Info and Collector is case-sensitive strings. Info object is tree structure no more than 3 depth, and Collector is table structure only pure geosicences dataset which is columns forms comprised of several json array.

As another shorthand notation, the term "GPJSON types" refers to two case-sensitive strings: " MainInfo ", " MainCollectors ", and the features types listed above.

A GPJSON text is a JSON text and consists of a single GPJSON object.

A GPJSON object represents a MainInfo, MainCollectors, or InfoTree and DataTable of Features.

A GPJSON object is a JSON object.

A InfoTree object has a member with the name "GeoMethod". The value of the member MUST be one of the GPJSON types which is a method of Geophysical field.

A InfoTree object MAY have a UnitsMap, which is a units list of DataTable keys. For exampe, Value0 : “celsius degree” pairs represent physical of values array of DataTable.

DataTable is numeric array data type of json.

Location is a no more than 3 numeric array to reprecentLocation point (X, Y) of local coordinate system in Geophysical field. Implementations SHOULD NOT extend Location arrays beyond 2 elements.

Geodesic is a no more than 3 numeric array to reprecentLocation point (X, Y, Z) of Geodetic Coordinate System，such as WGS84，The definition of the China Geodetic Coordinate System 2000. But the Coordinate ReferenceSystem (CRS) must be illustrated at CRS object. Implementations SHOULD NOT extend Geodesic arrays beyond 3 elements.

A GPJSON object MAY have other members

4. MainInfo Object

MainInfo is a tree structures to reprecent some Global Informations in this GPJson Document.

5. MainCollectors Object

MainInfo is a list structures to reprecent all data object which is consist of infoTree object and DataTable Object.

6. InfoTree Obect

InfoTree is a tree structures to reprecent Informations in a single Data Object. Some necessary information MUST be declared in this object, including listed Definitions above.

7. DataTable Object

DataTable is a table structures to reprecent raw data in a single Data Object. The content of DataTable object MUST be numeric array of json format. At this point, numberic array is in fact is Columns structure to declare rar data set of Geophyical field. Consequently, the ID of all array MUST have one-to-one correspondence between the array of DataTable.

8. Coordinate Reference System

The coordinate reference system for all GPJSON coordinates is a geographic coordinate reference system, using the World Geodetic System 1984 (WGS 84) [WGS84] datum, with longitude and latitude units of decimal degrees. This is equivalent to the coordinate reference system identified by the Open Geospatial Consortium (OGC) URN urn:ogc:def:crs:OGC::CRS84. An OPTIONAL third-position element SHALL be the height in meters above or below the WGS 84 reference

ellipsoid. In the absence of elevation values, applications sensitive to height or depth SHOULD interpret positions as being at local ground or sea level.

Note: the use of alternative coordinate reference systems was specified in [GJ2008], but it has been removed from this version of the specification because the use of different coordinate reference systems -- especially in the manner specified in [GJ2008] -- has proven to have interoperability issues. In general, GPJSON processing software is not expected to have access to coordinate reference system databases or to have network access to coordinate reference system transformation parameters. However, where all involved parties have a prior arrangement, alternative coordinate reference systems can be used without risk of data being misinterpreted.

9. Bounding Box

This section is adopted Bounding Box of GeoJson format.

A GeoJSON object MAY have a member named "bbox" to include information on the coordinate range for its Geometries, Features, or FeatureCollections. The value of the bbox member MUST be an array of length 2\*n where n is the number of dimensions represented in the contained geometries, with all axes of the most southwesterly point followed by all axes of the more northeasterly point. The axes order of a bbox follows the axes order of geometries.

The "bbox" values define shapes with edges that follow lines of constant longitude, latitude, and elevation .

Example of a 2D bbox member on a Feature:

|  |
| --- |
| "bbox": [-10.0, -10.0, 10.0, 10.0] |

Example of a 2D bbox member on

|  |
| --- |
| "bbox": [100.0, 0.0, 105.0, 1.0], |

Example of a 3D bbox member with a depth of 100 meters:

|  |
| --- |
| "bbox": [100.0, 0.0, -100.0, 105.0, 1.0, 0.0], |

9.1. The Connecting Lines

The four lines of the bounding box are defined fully within the coordinate reference system; that is, for a box bounded by the values "west", "south", "east", and "north", every point on the northernmost line can be expressed as

|  |
| --- |
| (lon, lat) = (west + (east - west) \* t, north)  with 0 <= t <= 1. |

5.2. The Antimeridian  
Consider a set of point Features within the Fiji archipelago, straddling the antimeridian between 16 degrees S and 20 degrees S. The southwest corner of the box containing these Features is at 20

degrees S and 177 degrees E, and the northwest corner is at 16 degrees S and 178 degrees W. The antimeridian-spanning GeoJSON bounding box for this FeatureCollection is

|  |
| --- |
| "bbox": [177.0, -20.0, -178.0, -16.0] |

and covers 5 degrees of longitude.

The complementary bounding box for the same latitude band, not crossing the antimeridian, is

|  |
| --- |
| "bbox": [-178.0, -20.0, 177.0, -16.0] |

and covers 355 degrees of longitude.

The latitude of the northeast corner is always greater than the latitude of the southwest corner, but bounding boxes that cross the antimeridian have a northeast corner longitude that is less than the

longitude of the southwest corner.

9.3. The Poles

A bounding box that contains the North Pole extends from a southwest corner of "minlat" degrees N, 180 degrees W to a northeast corner of 90 degrees N, 180 degrees E. Viewed on a globe, this bounding box approximates a spherical cap bounded by the "minlat" circle of latitude.

|  |
| --- |
| "bbox": [-180.0, minlat, 180.0, 90.0] |

A bounding box that contains the South Pole extends from a southwest corner of 90 degrees S, 180 degrees W to a northeast corner of "maxlat" degrees S, 180 degrees E.

|  |
| --- |
| "bbox": [-180.0, -90.0, 180.0, maxlat] |

A bounding box that just touches the North Pole and forms a slice of an approximate spherical cap when viewed on a globe extends from a southwest corner of "minlat" degrees N and "westlon" degrees E to a northeast corner of 90 degrees N and "eastlon" degrees E.

|  |
| --- |
| "bbox": [westlon, minlat, eastlon, 90.0] |

Similarly, a bounding box that just touches the South Pole and forms a slice of an approximate spherical cap when viewed on a globe has the following representation in GPJSON.

|  |
| --- |
| "bbox": [westlon, -90.0, eastlon, maxlat] |

Implementers MUST NOT use latitude values greater than 90 or less than -90 to imply an extent that is not a spherical cap.

10．UnitsMap object

The value of the optional " UnitsMap " member must be a string key/value that hints at the format used to represent physical units of DataTable object array. Suggested physical units are standardized, but others can be used.

11. URL object

If there ar URL key in TableTable object， the value must is a string to represented a raw data file on local machine or web.

12. Extending GPJSON

Members not described in this specification ("foreign members") MAY be used in a GPJSON document. Note that support for foreign members can vary across implementations, and no normative processing model for foreign members is defined. Accordingly, implementations that rely too heavily on the use of foreign members might experience reduced interoperability with other implementations.

10. Example

Most GPJson object can be specified using the following:

|  |
| --- |
| {  MainInfo : {  }  MainCollectors : [  {  InfoTree : {    },  DataTable : {    }  },  {  InfoTree : {    },  DataTable : {    }  },  {.....}  ]  } |

MainInfo and Info object represent basic information of recording/sounding by using json tree structure, the MainInfo object is common/global information and Info object is information of specific recording/sounding point. Collector represent recording/sounding raw data with table or columns structure.

|  |
| --- |
| {  "type" : "RawGeophysicalData",  "MainInfo" : {  "Method" : "TEM", // TEM geophysical Method  "Instrument" :" EAGeo\_TEM ", // acquisition from EAGEO-TEM instrument.  "Region" : "jiaozuo" , // work area  "Handler" : "aben lee" , // operator  "ExternalProgram" : " ", // option, for using external program  "arguments" : "....", // option, for external program arguments et c.  "crs" : { //  type: " urn:ogc:def:crs:OGC:1.3:CRS84",  properties: {  }  },  "...." : "...."  },  MainCollectors :: [  { " InfoTree ":{  "Location" : [1.0,2.0 ], // vector2D or vector3D for rcording point.  "Geodesic" : [Null,Null,Null], //option, geodetic coordinates in WGS84,  "..." : "..."  UnitsMap : { "Values0" : “ ” // refer to variables of NETCDF "Values1" : “ ”  …….  }  },  DataTable : {  “BCDTime ” : [ ], // Date Time array, BCD format  “Position” : [[x0,y0,z0], [x1,y1,z1] ], // X Y Z of Position  "Values0" : [3.0,2.2,...],  "Values1" : [1.0,3.1,...],  "URL" : "...." // for local file source or web data source  // for external geophysical data processing software or library.  }  },  {.....}  ]  } |

Note: DataTable is designed columns structure, the columns header is texted key of array. The array MUST [one-to-one correspondence](http://dict.youdao.com/w/one-to-one%20correspondence/#keyfrom=E2Ctranslation). At this point, The DataTable columns features as following viewers.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | BCDTime | Position | | | Values0 | Values1 |  |
| X | Y | Z |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Note： the ID is the number of arrays. However, in some cases, The BCDTime and Position is not necessary, since there are location or Geodesic of InfoTree structures

There are two kinds of GPJson format. one is of raw geophysical data, and the other if of processed geophysical data.

<<in progress>>

Additional information:

Magic number(s): n/a

File extension(s): .json, .gpjson

Macintosh file type code: n/a

Object Identifiers: n/a

Windows clipboard name: GPJSON

Macintosh uniform type identifier: public.gpjson conforms to public.json

Person to contact for further information: aben lee (aben.lee@foxmail.com)

Intended usage: COMMON

Restrictions on usage: none

Restrictions on usage: none

Author: aben lee

Change controller: Eageo Information & Exploration Technology Research Institution (EIET) from Henan Polytechnic University (hpu), P.R.China

**References:**

http://www.oschina.net/translate/geojson-spec#linked-crs

http://GeoJson.org

NETCDF format specification

**Appendix A.**