Open Geospatial Consortium

OGC Features and Geometries JSON (JSON-FG)

Clemens Portele, interactive instruments

08 September 2022

Overview JSON-FG

- Intentional limitations exist in GeoJSON:
  - Restricted to WGS 84 as Coordinate Reference System
  - Points, line strings and polygons – no support for solids or prisms
  - Supports spatial, but not temporal geometries
  - No feature type concept, no information about the schema
- Develop an OGC Features and Geometries JSON standard (JSON-FG) addressing these limitations
  - Avoid edge cases, focus on capabilities that are useful for many spatial experts
  - Specify as a superset of GeoJSON: Valid JSON-FG is also valid GeoJSON
- Status: Initial development and testing completed, a draft for wider testing should be available soon

GeoJSON is the starting point

- JSON
  - { ... } is an object with key/value pairs (members)
  - [...] is an array
- GeoJSON
  - A feature collection is an object; predefined keys:
    - "type" – always "FeatureCollection"
    - "features" – an array of features
  - A feature is an object; predefined keys:
    - "type" – always "Feature"
    - "id" – an optional identifier
    - "geometry" – a Simple Feature geometry (Point, LineString, etc.) in WGS 84 longitude, latitude and optional ellipsoidal height
    - "properties" – an object that can contain feature properties, GeoJSON does not place any constraints on the contents

```
{
   "type": "FeatureCollection",
   "features": [
      {
         "type": "Feature",
         "id": "DENW19AL0000giv5BL",
         "geometry": {
            "type": "Point",
            "coordinates": [ 8.7092045, 51.503528 ]
         },
         "properties": {
            "address": "..."
            "lastChange": "2014-04-24T10:50:18Z",
            "built": "2012-03",
            ...
         }
      },
      ...
   ]
}
```

Encoding temporal information

- GeoJSON supports spatial geometries (‘geometry’)
- Features are often associated with temporal information, too
- OGC API Features supports not only spatial, but also temporal filtering (`datetime` parameter)
- JSON-FG adds support for the most common case
  - associating a feature with a single temporal instant or interval in the Gregorian calendar
  - main use case is filtering (time slider) or display without the need to understand the feature schema
  - leveraging RFC 3339 and ISO 8601
- No constraints how this primary temporal geometry is derived from the feature properties

```
{
   "type": "Feature",
   ...,
   "time": {
      "interval": [ "2014-04-24T10:50:18Z", ".." ]
   },
   ...,
   "properties": {
      "lastChange": "2014-04-24T10:50:18Z",
      "built": "2012-03",
      ...
   }
}
```

top-level member "time"

Encoding a spatial geometry (1/3)

- GeoJSON supports Simple Features geometries (2D or 2.5D points, line strings, polygons or aggregations of them) in WGS 84
- A geometry that meet these constraints will always be in the "geometry" member from GeoJSON

```
{
   ...,
   "geometry": {
      "type": "Polygon",
      "coordinates": [
         [
            [ 8.70920456, 51.50352856, 100 ], ...
            [ 8.70920456, 51.50352856, 100 ]
         ]
      ]
   },
   ...
}
```

Encoding a spatial geometry (2/3)

- Other geometries are added in a top-level member "place"
  - Geometry is a solid or a prism (extruded polygon)
    - Support for arcs and circles under discussion
  - Geometry is in another coordinate reference system (CRS)
    - The CRS is declared in "coordRefSys"
    - See `crs` query parameter from OGC API Features

top-level member "coordRefSys" declares the CRS in the "place" geometry

```
{
   ...,
   "coordRefSys": "http://www.opengis.net/def/crs/
                   EPSG/0/5555",
   "place": {
      "type": "Polyhedron",
      "coordinates": [
         [
            [
               [ 479816.67, 5705861.672, 100 ], ...
               [ 479816.67, 5705861.672, 100 ]
            ]
         ], ...
      ]
   },
   ...
}
```

top-level member "place"

Encoding a spatial geometry (3/3)

- To support GeoJSON readers a fallback geometry can be added in the GeoJSON "geometry" member

```
Accept: application/vnd.ogc.fg+json;compatibility=geojson,
application/vnd.ogc.fg+json; q=0.9, application/geo+json; q=0.8

{
   ...,
   "coordRefSys": "http://www.opengis.net/def/crs/EPSG/0/
                        5555",
   "place": {
      "type": "Polyhedron",
      "coordinates": [
         [
            [
               [ 479816.67, 5705861.672, 100 ], ...
               [ 479816.67, 5705861.672, 100 ]
            ]
         ], ...
      ]
   },
   "geometry": {
      "type": "Polygon",
      "coordinates": [
         [
            [ 8.709204563652449, 51.50352856284526, 100 ], ...
            [ 8.709204563652449, 51.50352856284526, 100 ]
         ]
      ]
   },
   ...
}
```

a valid GeoJSON geometry in "geometry"

Identifying the feature type(s)

- Features are often categorized by type
  - typically one feature type, but multiple feature types are supported, too
  - in a Features API the features are often grouped by type into collections
- GIS clients often depend on knowledge about the feature type
  - example: to associate a style to render the feature on a map
- GeoJSON has no concept of feature types or feature schemas

a token for filtering

```
{
   "type": "Feature",
   "id": "DENW19AL0000giv5BL",
   "featureType": "app:building",

   ...

   "links": [
      {
         "href": "https://inspire.ec.europa.eu/
                 featureconcept/Building",
         "rel": "type",
         "title": "This feature is of type 'building'"
      }
   ],
   ...
}
```

in addition, a link to the semantic type definition in some registry, if available

Identifying the schema(s)

- Language: JSON Schema
- Clients can use schemas to validate the JSON document or to derive additional information about the content
- Follow the JSON Schema guidance:
  - *It is RECOMMENDED that instances described by a schema provide a link to a downloadable JSON Schema using the link relation "describedby".*
- Determine that an instance is a GeoJSON / JSON-FG feature though the canonical URIs of the schemas

```
{
   ...,
   "links": [
      {
         "href": "https://ogc-api.nrw.de/lika/v1/
                  collections/gebaeude_bauwerk/schema",
         "rel": "describedby",
         "type": "application/schema+json",
         "title": "JSON Schema of this document"
      },
      {
         "href":"http://schemas.opengis.net/tbd/
                Feature.json",
         "rel":"describedby",
         "type":"application/schema+json",
         "title":"This document is a JSON-FG Feature"
      },
      {
         "href":"https://geojson.org/schema/
                Feature.json",
         "rel":"describedby",
         "type":"application/schema+json",
         "title":"This document is a GeoJSON Feature"
      }
   ],
   ...
}
```

links to all schemas that the document conforms to

Declaring information in the feature collection

- To simplify processing by clients
- For homogenous feature collections, it is sufficient to include the feature type information once – in the feature collection
- If all features in the feature collection have geometries of the same dimension, this can be declared, too
  - 0: points
  - 1: curves
  - 2: surfaces
  - 3: solids
  - no value: unknown or mixed
- Declare a default coordinate reference system

```
{
   "type": "FeatureCollection",
   "featureType": "app:building",
   "geometryDimension": 2,
   "coordRefSys": "http://www.opengis.net/def/crs/
                   EPSG/0/5555",
   "features": [
      ...
   ]
}
```

More information

- Draft:
  - https://docs.ogc.org/DRAFTS/21-045.html
- GitHub repository:
  - https://github.com/opengeospatial/ogc-feat-geo-json
- Implementations:
  - https://github.com/opengeospatial/ogc-feat-geo-json/tree/main/implementations

- Contact us:
  - Panagiotis (Peter) A. Vretanos (CubeWerx Inc.)
    - pvretano [at] cubewerx.com
  - Clemens Portele (interactive instruments)
    - portele [at] interactive-instruments.de

Thank You

Community

500+ International Members

110+ Member Meetings

60+ Alliance and Liaison partners

50+ Standards Working Groups

45+ Domain Working Groups

25+ Years of Not for Profit Work

10+ Regional and Country Forums

Innovation

120+ Innovation Initiatives

380+ Technical reports

Quarterly Tech Trends monitoring

Standards

65+ Adopted Standards

300+ products with 1000+ certified implementations

1,700,000+ Operational Data Sets

Using OGC Standards