GeoJSON

Info Sheet

Geospatial Extended JSON :

GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents.
GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984, and units of decimal degrees.

Typical GeoJSON Structure

GeoJSON files represent a layer of geometric features, usually a ***FeatureCollection***. The FeatureCollection has an array of ***features***. A Feature contains the following attributes:

- type
- geometry
- properties

```
{ "type": "FeatureCollection",
  "features":
  [
   { "type": "Feature",
     "geometry": { "type": "Point",  "coordinates": [102.0, 0.5] },
     "properties": { "prop0": "value0" }
    },
   { "type": "Feature",
     "geometry": {
        "type": "LineString",
        "coordinates": [[102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0] ]
      },
     "properties": { "prop0": "value0", "prop1": 0.0 }
    },
    { "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates":
           [[
            [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0]
           ]]
       },
      "properties": { "prop0": "value0", "prop1": {"this": "that"} }
    }
   ]
}
```

Basic Vector Data Types

The geometry ***type***s supported by GeoJSON include vector geometries:

- Point
- LineString
- Polygon

These geometric data elements are defined by coordinates, a single coordinate is a 2- or 3-element list:

- `[longitude, latitude]`
- `[longitude, latitude, elevation]`

Geometry is the planar vector type, where the spatial reference determines the underlying coordinate system as well as the unit of measure. Geometries are JSON objects with *type* and *coordinates.*
Example of Geometries:

```
• { "type": "Point", "coordinates": [0, 0] }
• { "type": "LineString", "coordinates": [[0, 0], [10, 10]] }
• { "type": "Polygon",
     "coordinates": [[
       [0, 0], [10, 10], [10, 0], [0, 0]
      ]]
  }
```

Note: As you have learned, polygons are defined as an outer closed LineString (perimeter) and zero or more inner closed LineStrings (holes).

Each of these geometries can be an element of a *geometries* list, forming a “Multi” version of the geometry in a “GeometryCollection” type.

References and Related Technologies

- General: https://en.wikipedia.org/wiki/GeoJSON
- Technical Specification: https://tools.ietf.org/html/rfc7946
- Python – GeoPandas: http://geopandas.org/index.html
- PostGIS – ST_AsGeoJSON(): http://www.postgis.net/docs/ST_AsGeoJSON.html