

CityJSON

Exemple simple d'un bâtiment :

visionnable sur <https://ninja.cityjson.org/>

- attributs de hauteur du bâtiment + informations

```
1  {
2    "CityObjects": {
3      "1": {
4        "attributes": {
5          "ArrDissolve-LoD12.global_elevation_50p": 6.384429931640625,
6          "ArrDissolve-LoD12.global_elevation_70p": 6.689910888671875,
7          "ArrDissolve-LoD12.global_elevation_max": 7.1895751953125,
8          "ArrDissolve-LoD12.global_elevation_min": 0.05712890625,
9          "DataCoverageCalc.data_coverage": 0.9169994592666626,
10         "LASInPolygons.ground_elevations": 0,
11         "OGRLoader.annee": "2021",
12         "OGRLoader.geo_batime": "0401693",
13         "OGRLoader.geo_dur": "01",
14         "OGRLoader.lot": "112",
15         "OGRLoader.object_fid": "Objet_15962",
16         "OGRLoader.ogc_fid": 1693,
17         "OGRLoader.tex": "",
18         "PC2MeshQuality-LoD12.mesh_error_f": 0.5470260381698608,
19         "PC2MeshQuality-LoD13.mesh_error_f": 0.5470260381698608,
20         "PC2MeshQuality-LoD22.mesh_error_f": 0.02626117132604122,
21         "Validator-LoD12.errors": "[]",
22         "Validator-LoD13.errors": "[]",
23         "Validator-LoD22.errors": "[]"
24       },
25     },
26   },
27 }
```

Des lignes 25 à 43 : LoD 0 définit par :

- geographicalExtent : emprise2D de la "geometry"
- "boundaries" : liste d'index qui va piocher dans tous les "vertices", une liste définie plus bas qui a tous les sommets en coordonnées x, y, z dans une base locale. Le premier sommet de cette liste est alors [0, 0, 0].
- lod et type

```
25   "children": [
26     "1-0"
27   ],
28   "geographicalExtent": [
29     923357.64,
30     6307320.449999962,
31     0.0,
32     923370.9100004578,
33     6307332.540000115,
34     0.0
35   ],
36   "geometry": [
37     {
38       "boundaries": [[[0,1,2,3,4,5]]],
39       "lod": "0",
40       "type": "MultiSurface"
41     }
42   ],
43   "type": "Building"
44 },
45 "1-0": {
```

Suite "1-0" :

- geometry définit autant d'enfants que de lod. Ici 1.2 1.3 et 2.2
- on retrouve différentes boundaries selon le lod, on observe que certains points sont utiles à la construction de plusieurs lod, là où les détails du 2.2 ne sont qu'utiles pour ce lod

```

45     "1-0": {
46         "attributes": {},
47         "geometry": [
48             {
49                 "boundaries": [[[6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]], [[23, 11, 10, 24]]]
50                 "lod": "1.2",
51                 "semantics": {
52                     "surfaces": [
53                         {"type": "GroundSurface"},
54                         {"type": "RoofSurface"},
55                         {"on_footprint_edge": true, "type": "WallSurface"},
56                         {"on_footprint_edge": false, "type": "WallSurface"}
57                     ],
58                     "values": [[0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1]]
59                 },
60                 "type": "Solid"
61             },

```

- ensuite, metadata qui contient empreinte etc
- transform et translate pour placer les vertices au bon endroit et à la bonne taille
- les vertices

```

95     "metadata": {
96         "geographicalExtent": [
97             923357.64,
98             6307320.449999962,
99             0.0,
100            923370.9100004578,
101            6307332.540000115,
102            365.1239929199219
103        ],
104        "identifier": "",
105        "referenceDate": "",
106        "referenceSystem": "EPSG::2154",
107        "title": "3D model generated by Geoflow"
108    },
109    "transform": {
110        "scale": [
111            0.001,
112            0.001,
113            0.001
114        ],
115        "translate": [
116            923357.64,
117            6307326.91,
118            0.0
119        ]
120    },
121    "type": "CityJSON",
122    "version": "1.1",
123    "vertices": [[0, 0, 0], [4540, -6460, 0], [9100, -3160, 0], [9870, -4070, 0], [13270, -1710, 0], [7840, 5630, 0], [910
124    ]

```