

Unofficial DfT Connectivity Tool

API Contract (v1)

Public-facing contract for clients that want to query connectivity data or request downloadable exports.

Base URL:

None

<https://labs.podaris.com/api/connectivity-metric-extractor>

[Overview](#)

[CORS](#)

[Health](#)

[GET /health](#)

[Point Lookup](#)

[GET /v1/connectivity](#)

[API Key Issuance](#)

[POST /access/request-key](#)

[Authenticated API Export](#)

[POST /v1/export](#)

[Request body](#)

[Success response](#)

[GET /v1/export/:jobId](#)

[GET /v1/export/:jobId/download](#)

[Validation errors](#)

[Output Schema](#)

[GeoJSON](#)

[CSV](#)

Overview

Supported workflows:

- API key request: `POST /access/request-key`
- authenticated connectivity lookup: `GET /v1/connectivity`
- authenticated export request: `POST /v1/export`
- authenticated export status: `GET /v1/export/:jobId`
- authenticated export download: `GET /v1/export/:jobId/download`

Supported geography levels:

- `grid100`
- `oa`
- `lsoa`
- `lad`
- `rgn`

Supported export formats:

- `geojson` (default)
- `csv`

CORS

The API supports:

- `GET`
- `POST`
- `OPTIONS`

Cross-origin requests are allowed with:

```
None
Access-Control-Allow-Origin: *
```

Health

`GET /health`

Returns a simple health response.

Example response:

```
JSON
{
  "ok": true,
  "ts": "2026-05-06T20:39:04.119Z"
}
```

Point Lookup

GET /v1/connectivity

Authenticated API endpoint. Requires:

```
None
Authorization: Bearer dft_live_...
```

Resolves a single latitude/longitude to a geography and optionally returns metrics.

Query parameters:

- `lat` required, number
- `lon` required, number
- `level` optional, default `grid100`
- `include_metrics` optional, `true` or `false`, default `true`

Example:

```
Shell
curl -sS
"https://labs.podaris.com/api/connectivity-metric-extractor/v1/connectivity?lat=51.5074&lon=-0.1278&level=oa" \
-H "Authorization: Bearer dft_live_..."
```

Example response:

```
JSON
{
  "input": {
    "lat": 51.5074,
    "lon": -0.1278,
    "level": "oa"
  },
  "resolved": {
```

```
"gridId": "530000_180000",
"bngE": 530000,
"bngN": 180000,
"codes": {
  "grid100": "530000_180000",
  "oa": "E00000001",
  "lsoa": "E01000001",
  "lad": "E09000001",
  "rgn": "E12000007"
},
"metrics": {
  "Overall": 52.8
}
}
```

Possible errors:

- 400 invalid_lat_lon
- 400 invalid_level
- 404 grid_id_not_found

API Key Issuance

POST /access/request-key

Requests an API key to be emailed to the given address.

Request body:

```
JSON
{
  "email": "you@example.com"
}
```

Success response:

```
JSON
{
  "status": "emailed"
}
```

Authenticated API Export

POST /v1/export

Authenticated export endpoint for API consumers.

Required header:

```
None
Authorization: Bearer dft_live_...
```

Differences from browser export:

- `email` is not accepted in the request body
- the notification email is derived from the API key owner for analytics/auditing only
- no ready or failure email is sent for authenticated API exports
- status and download are available under `/v1/export/...`

Request body

Required fields:

- `level` one of `grid100|oa|lsoa|lad|rgn`
- `geometry`

Optional fields:

- `format` one of `geojson|csv`, default `geojson`
- `include_metrics` boolean, default `true`
- `source` optional client/source label
- `context` optional client/context label

Supported `geometry` values:

1. `bbox`

```
JSON
{
  "type": "bbox",
  "bbox": [-0.1330, 51.5030, -0.1185, 51.5120]
}
```

2. GeoJSON Polygon

3. GeoJSON MultiPolygon

Example:

```
Shell
curl -sS -X POST
"https://labs.podaris.com/api/connectivity-metric-extractor/v1/export" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer dft_live_xxx" \
-d '{
  "level": "oa",
  "format": "geojson",
  "geometry": {
    "type": "bbox",
    "bbox": [-0.1330, 51.5030, -0.1185, 51.5120]
  }
}'
```

Success response

Status:

```
None
202 Accepted
```

Body:

```
JSON
{
  "job_id": "879b4018-ffb2-481b-805e-7982b2396227",
  "status": "queued",
  "async": true,
  "source": "api"
}
```

GET /v1/export/:jobId

Returns authenticated job status for jobs owned by the presented API key.

Example response:

```
JSON
{
  "id": "879b4018-ffb2-481b-805e-7982b2396227",
```

```
{
  "status": "done",
  "created_at": "2026-05-07T12:00:00.000Z",
  "started_at": "2026-05-07T12:00:02.000Z",
  "completed_at": "2026-05-07T12:00:10.000Z",
  "level": "oa",
  "format": "geojson",
  "include_metrics": true,
  "point_count": 1278,
  "source": "api",
  "error": null,
  "output_key": "exports/879b4018-ffb2-481b-805e-7982b2396227.geojson",
  "email": "you@example.com"
}
```

GET /v1/export/:jobId/download

Downloads the generated file for jobs owned by the presented API key. No browser token is required on this route.

Validation errors

Possible 400 errors include:

- invalid_json
- invalid_level
- email_required
- geometry_required
- invalid_bbox
- unsupported_geometry_type
- geometry_too_large
- area_too_large

Example oversized geometry response:

```
JSON
{
  "error": "geometry_too_large",
  "message": "Export geometry is too detailed to queue (3160992 bytes). Please simplify the shape or upload a smaller area.",
  "geometry_bytes": 3160992,
  "limit_bytes": 98304
}
```

Example oversized area response:

JSON

```
{
  "error": "area_too_large",
  "message": "Export area (41000 km²) exceeds the 40000 km² limit for oa exports.",
  "area_km2": 41000,
  "limit_km2": 40000
}
```

Current export area limits:

- grid100: 5000 km²
- oa: 40000 km²
- lsoa: 90000 km²
- lad: no area cap
- rgn: no area cap

Output Schema

Properties vary by level. Only fields that are meaningful at a given level are included — fields that would always be null are omitted.

GeoJSON

GeoJSON exports return a `FeatureCollection`. Each feature's geometry is the boundary polygon of the geography. Metrics, when included, are flattened as top-level properties alongside the geographic fields (no nested `metrics` object).

Properties present per level:

Field	grid100	oa	lsoa	lad	rgn
lat, lon	✓	—	—	—	—
grid_id	✓	—	—	—	—
oa	✓	✓	—	—	—
lsoa	✓	✓	✓	—	—
lad	✓	✓	✓	✓	—
rgn	✓	✓	✓	✓	✓
level	✓	✓	✓	✓	✓

Field	grid100	oa	lsoa	lad	rgn
level_code	✓	✓	✓	✓	✓
metrics	if requested	if requested	if requested	if requested	if requested

For **grid100**, the feature geometry is a 100m × 100m square polygon. **lat/lon** are the centroid of the cell and **grid_id** is the BNG easting/northing of the cell's SW corner (e.g. 530050_180050).

Example **grid100** feature properties:

```
JSON
{
  "lat": 51.5074,
  "lon": -0.1278,
  "grid_id": "530000_180000",
  "oa": "E00000001",
  "lsoa": "E01000001",
  "lad": "E09000001",
  "rgn": "E12000007",
  "level": "grid100",
  "level_code": "530000_180000",
  "Overall": 52.8
}
```

Example **oa** feature properties:

```
JSON
{
  "oa": "E00000001",
  "lsoa": "E01000001",
  "lad": "E09000001",
  "rgn": "E12000007",
  "level": "oa",
  "level_code": "E00000001",
  "Overall": 52.8
}
```

Example **lad** feature properties:

JSON

```
{
  "lad": "E09000001",
  "rgn": "E12000007",
  "level": "lad",
  "level_code": "E09000001",
  "Overall": 52.8
}
```

CSV

CSV exports follow the same level-dependent schema. Columns present per level mirror the table above. Metrics, when included, are one column per metric name.