



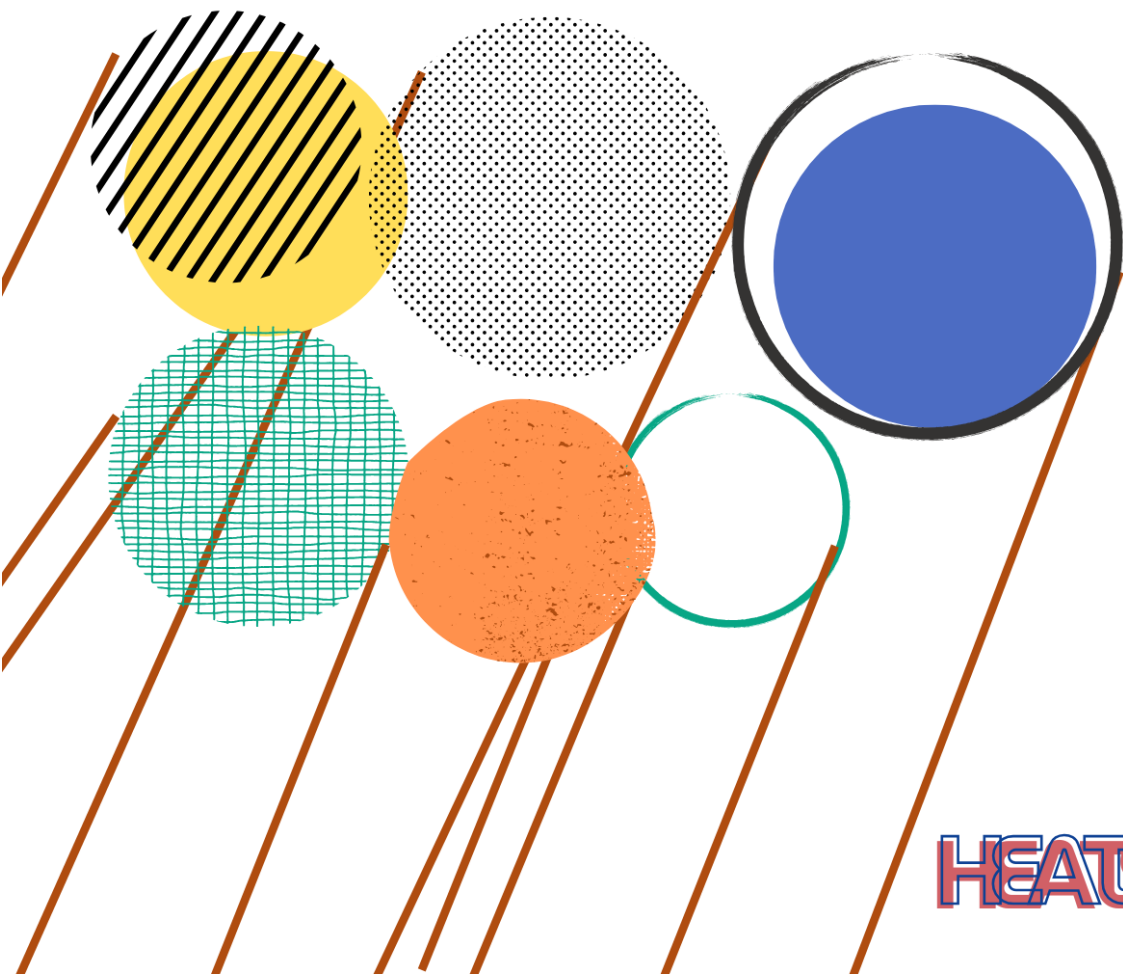
HEATSAFE

Interreg
Euro-MED



Co-funded by
the European Union

Data Guidelines for HEATSAFE tool



HEATSAFE



Table of Contents

Building Your Own “HEATSAFE”-Style Area Dataset.....	2
1.File-level wrapper	2
2.Per-feature structure	3
3. The properties object — field-by-field reference	3
3.1 Row identifiers.....	3
3.2 Wet-bulb temperature (heat-stress) metrics.....	4
3.3 First geographic/statistical join (.x suffix — census-cell identity).....	4
3.4 Demographics (percentages, 0–100, per statistical cell).....	4
3.5 Second geographic/statistical join (.y suffix — secondary dataset match)	6
3.6 Household energy use — cooking, heating, hot water (percentages, 0–100).....	6
3.7 Building insulation & type (percentages, 0–100).....	7
3.8 Misc. / derived indicators.....	7
3.9 Classification codes	8
4. Null-handling convention.....	8
5. Checklist for building MYAREA.js	9
6. Full example (one representative feature, all keys populated).....	9

Building Your Own “HEATSAFE”-Style Area Dataset

For the user to add local area data into the HEATSAFE interactive platform, a specifically formatted **MYAREA.js** file needs to be created. This is a small piece of JavaScript that registers a **GeoJSON FeatureCollection** into a global lookup object, keyed by an area code. Each “feature” is one small spatial unit (a building block / parcel) carrying two things: a **geometry** (its polygon shape) and a bundle of **properties** — climate, demographic, housing, and vulnerability-index data for that unit.

This guide breaks down the structure and walks through recreating it for a specific area.

1.File-level wrapper

The whole file is a single JavaScript statement: a call to a global HEATSAFE registry object, passing an area code and a GeoJSON FeatureCollection as arguments. There is no `export`, `var`, or module syntax — the file's only job is to register data as a side effect of being loaded (e.g. via a `<script src="MYAREA.js">` tag).

```
HEATSAFE.put("MYAREA", {"type": "FeatureCollection", "features": [
  { "type": "Feature", "properties": { ... }, "geometry": { ... } },
  { "type": "Feature", "properties": { ... }, "geometry": { ... } },
```





```
...
{ "type": "Feature", "properties": { ... }, "geometry": { ... } }
]
});
```

Key points:

- **First argument to `HEATSAFE.put`:** a short uppercase area code as a string ("MYAREA"). This is the key other code will use to look up the dataset, so it must be unique across all area files loaded into the app and should match the filename stem.
- **Second argument:** a standard GeoJSON FeatureCollection object with exactly two top-level keys: "type": "FeatureCollection" and "features".
- Each feature is written on its own single line (no pretty-printing/indentation inside the feature), features are comma-separated, and the array is closed with `]` on its own line followed by `});` on the next line. There's no trailing comma after the last feature.
- Numbers are plain JSON numbers (floats print with full double precision, e.g. 0.6666666666666663), strings are double-quoted, and missing values are JSON `null` (not "NA", not undefined, not omitted keys).

2. Per-feature structure

Every entry in `features` is a GeoJSON Feature with exactly three keys, always in this order:

```
{ "type": "Feature", "properties": { ... 74 keys ... }, "geometry": { "type":
"MultiPolygon", "coordinates": [ ... ] } }
```

"type": always the literal string "Feature".

- "properties": a flat object of 74 attributes (see §3). Same 74 keys, in the same order, on every feature — including when the value is `null`.
- "geometry": always "type": "MultiPolygon" in this file, even for what is conceptually a single polygon (a MultiPolygon with one outer ring). `coordinates` is the standard GeoJSON MultiPolygon nesting: `[ polygon, polygon, ... ]` where each polygon is `[ ring, ring, ... ]` and each ring is `[ [lon, lat], [lon, lat], ... ]` (first and last point equal, i.e. the ring is closed). Coordinates are `[ longitude, latitude ]` decimal degrees (WGS84).

3. The properties object — field-by-field reference

The 74 property keys fall into logical groups. Order matters for consistency with existing files, even though GeoJSON consumers don't require it.

3.1 Row identifiers

Key	Type	Notes
-----	------	-------





Key	Type	Notes
fid	int	1-based row number, matches feature's position in the array. Unique per file.
ID	int	Identical to fid in this file — keep them equal.

3.2 Wet-bulb temperature (heat-stress) metrics

Key	Type	Notes
mean_max_wbt_14days	float	Mean of daily max wet-bulb temperature over a 14-day window, °C.
norm_mean_max_wbt	float	mean_max_wbt_14days normalized to 0–1 (min–max scaled across the area).
max_avg_wbt	float	Max of daily average wet-bulb temperature, °C.
norm_max_avg_wbt	float	max_avg_wbt normalized to 0–1.

3.3 First geographic/statistical join (.x suffix — census-cell identity)

Key	Type	Notes
ESYECODE.x	float	Cell code, stored as a float (e.g. 5.0).
OIKISMOS.x	string	Settlement/statistical-unit name, formatted <Area>_OT_<CODE2022.x>.
CODE2022.x	string	Census tract code, as a string of digits.
NMUN.x	string	Municipality name (English), e.g. "Agioli Anargyroi Municipality".

3.4 Demographics (percentages, 0–100, per statistical cell)

Key	Type	Notes
-----	------	-------



HEATS SAFE

Interreg
Euro-MEDCo-funded by
the European Union

Key	Type	Notes
pop_tot	float	Total population count of the cell (not a percentage).
under5	float	% of population under 5 (verify scale for your source data).
over65	float	% of population over 65.
males	float	% male. males + females sums to 100.0 on every row.
females	float	% female.
unemp	float	Unemployment indicator/percentage.
employed	float	% employed.
retired	float	% retired.
b_75	float	% of residents born before 1975 (age-cohort breakdown).
b75_80	float	% born 1975–1980.
b80_85	float	% born 1980–1985.
b85_90	float	% born 1985–1990.
b90_95	float	% born 1990–1995.
b95_00	float	% born 1995–2000.
b00_05	float	% born 2000–2005.
b05_10	float	% born 2005–2010.
b10_15	float	% born 2010–2015.
b15_20	float	% born 2015–2020.

The ten b_* cohort fields sum to ~100 per row (0 when the cell has no age data).



3.5 Second geographic/statistical join (.y suffix — secondary dataset match)

Key	Type	Notes
ESYECODE.y	float	Same shape as ESYECODE.x, from a second source table.
OIKISMOS.y	string	Same shape as OIKISMOS.x.
CODE2022.y	string	Same shape as CODE2022.x.
NMUN.y	string	Same shape as NMUN.x.

3.6 Household energy use — cooking, heating, hot water (percentages, 0–100)

Each of the three household-energy topics (cooking, heating, hotwater) has the same six source categories. Naming pattern: <Source>_<topic>:

Key	Type	Notes
Electricity_<topic>	float	e.g. Electricity_cooking, Electricity_heating, Electricity_hotwater
Natural_gas_<topic>	float	
Oil_<topic>	float	
Solar_energy_<topic>	float	
Biomass_<topic>	float	
Other_or_none_<topic>	float	

i.e. 18 keys total: the six sources above × the three topic suffixes (_cooking, _heating, _hotwater). All floats, percentages that sum to ~100 within each topic when data exists (0 across the board when the cell has no survey data).



3.7 Building insulation & type (percentages, 0–100)

Key	Type	Notes
Double_glass	float	% of dwellings with double glazing.
External_wall	float	% with external wall insulation.
External_insulation	float	% with (other) external insulation.
Other_insulation	float	% with another insulation type.
Two_or_more	float	% with two or more insulation measures.
No_insulation	float	% with no insulation.
Single_house	float	% single-family houses.
Double_house	float	% semi-detached/duplex houses.
Multi_house	float	% multi-family/apartment buildings.

3.8 Misc. / derived indicators

Key	Type	Notes
year	float or null	Survey/data year, e.g. 2020.0. null when unknown for that cell.
ac_pen	float or null	Air-conditioning penetration rate (0–1 or %; all 0.0 in this sample).
green_frac	float or null	Fraction of green space in the cell, 0–1.
grn_f_1	float or null	Secondary green-space fraction/metric, 0–~11 in this sample (check source units).
be75_80	float	Binary-looking flag (0.0 or 1.0) related to the b75_80 cohort/building era.



Key	Type	Notes
infra_assets	float or null	Infrastructure/asset adequacy score, observed values {0, 1/3, 2/3, 1} → looks like a count-of-3 ratio.
thermal_jus_ok	float	Thermal-justice adequacy score, observed values {0.0, 0.5, 1.0}.
health	float or null	Health-vulnerability flag/score; mostly 0.0, sometimes null.
edu_work_stds	float	Education/work standards flag, {0.0, 1.0}.
SCP_sel_th_okay_va_bene	float or null	Social/climate-plan composite sub-score, 0–1 range.
SCP_pop	float or null	Population count relevant to the SCP score (not a percentage).
pop_dens	float	Population density (people per km ² , large values e.g. 3,000–55,000+).

3.9 Classification codes

Key	Type	Notes
wbt_class	string	Wet-bulb-temperature class, one of "1", "2", "3", "4" (never null). Stored as a string, not a number.
scp_class	string or null	Social/climate-plan class, one of "1"–"4", or null when SCP_* inputs are unavailable.
bi_class	string	Derived field — see below.

bi_class is derived, not raw data — built as: `(scp_class ?? "NA") + "-" + wbt_class`.
 Examples: `scp_class="4", wbt_class="1" → bi_class="4-1"`; `scp_class=null, wbt_class="1" → bi_class="NA-1"`. Compute this last, after the two source classes are finalized, using exactly this concatenation order (scp first, wbt second).

4. Null-handling convention

- Numeric fields that are genuinely unknown/unavailable → JSON `null`.





- The .y join fields use "0" (string) / 0.0 (float) as their "no match" sentinel instead of null — replicate this exactly for consistency with existing consumers of the data, even though it's a slightly unusual convention.
- Do not omit keys for missing values — every feature must have all 74 property keys present, with null where appropriate, so downstream code can rely on a fixed schema.

5. Checklist for building MYAREA.js

1. Prepare one row of source data per spatial cell (statistical unit / grid cell) for MYAREA, with all 74 attributes listed above.
2. For each row, build a GeoJSON Feature: type, properties (all 74 keys, correct types, null/"0" conventions per §4), geometry (MultiPolygon, [lon, lat] closed rings).
3. Compute bi_class last: (scp_class if not null else "NA") + "-" + wbt_class.
4. Assign fid/ID as 1-based row numbers matching array order.
5. Wrap all features in {"type": "FeatureCollection", "features": [...]}.
6. Wrap that in HEATS SAFE.put("MYAREA", { ... }); and save as MYAREA.js, one feature per line, no trailing comma after the last feature.
7. Sanity-check: males + females == 100 per row where populated, b_* cohort fields sum to ~100, wbt_class/scp_class are strings not numbers, and every feature has exactly 74 property keys in the same order as AKK.js.

6. Full example (one representative feature, all keys populated)

```
{ "type": "Feature", "properties": { "fid": 8, "ID": 8, "mean_max_wbt_14days": 23.051005499703543,
  "norm_mean_max_wbt": 0.08851932556082778, "max_avg_wbt": 22.221235354741413,
  "norm_max_avg_wbt": 0.09114123209106789, "ESYECODE.x": 13.0,
  "OIKISMOS.x": "AgAnargyroi_OT_4703010101", "CODE2022.x": "4703010101",
  "NMUN.x": "Agiol Anargyroi Municipality", "pop_tot": 139.0, "under5": 1.0, "over65": 0.0,
  "males": 48.9, "females": 51.1, "unemp": 1.0, "employed": 45.3, "retired": 14.4,
  "b_75": 2.9, "b75_80": 44.3, "b80_85": 12.9, "b85_90": 14.3, "b90_95": 15.7,
  "b95_00": 4.3, "b00_05": 5.7, "b05_10": 0.0, "b10_15": 0.0, "b15_20": 0.0,
  "ESYECODE.y": 0.0, "OIKISMOS.y": "0", "CODE2022.y": "0", "NMUN.y": "0",
  "Electricity_cooking": 0.0, "Natural_gas_cooking": 0.0, "Oil_cooking": 0.0,
  "Solar_energy_cooking": 0.0, "Biomass_cooking": 0.0, "Other_or_none_cooking": 0.0,
  "Electricity_heating": 0.0, "Natural_gas_heating": 0.0, "Oil_heating": 0.0,
  "Solar_energy_heating": 0.0, "Biomass_heating": 0.0, "Other_or_none_heating": 0.0,
  "Electricity_hotwater": 0.0, "Natural_gas_hotwater": 0.0, "Oil_hotwater": 0.0,
  "Solar_energy_hotwater": 0.0, "Biomass_hotwater": 0.0, "Other_or_none_hotwater": 0.0,
```



HEATS SAFE

Interreg
Euro-MED



Co-funded by
the European Union

```
"Double_glass": 0.0, "External_wall": 0.0, "External_insulation": 0.0,  
"Other_insulation": 0.0, "Two_or_more": 0.0, "No_insulation": 0.0, "Single_house": 0.0,  
"Double_house": 0.0, "Multi_house": 0.0, "year": 2020.0, "ac_pen": 0.0,  
"green_frac": 1.0, "grn_f_1": 0.0, "be75_80": 1.0, "infra_assets": 0.6666666666666666,  
"thermal_jus_ok": 0.5, "health": 0.0, "edu_work_stds": 1.0,  
"SCP_sel_th_okay_va_bene": 0.6333333333333333, "SCP_pop": 88.03333333333333,  
"pop_dens": 15764.203974439282, "wbt_class": "1", "scp_class": "4", "bi_class": "4-1" },  
"geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 23.7169, 38.0151 ],  
[ 23.7166, 38.0152 ], [ 23.7169, 38.0151 ] ] ] ] }
```