Datenmanagement Haecksenkarte

Daten

- momentan als json gespeichert, Format:

```
{
            "type": "Feature",
            "properties": {
                "fid": 1,
                "typ": "space",
                "count_fictive": 18,
                "space_name": "name"
                // more attributes
            },
            "geometry": {
                "type": "Point",
                "coordinates": [
                    9.0,
                    52.0,
                ],
            }
        }
```

- Repository HaeckMapData (private): https://gitlab.com/haeckmap/HaeckMapData
- neue Tabellen:
  - create table spaces (id number, type text, location text, x number, y number);
  - create table properties (id number, private boolean, type text, key text, value text);
  - create table options (key text, option text);
- dadurch können beliebige neue Eigenschaften hinzugefügt werden, unterstützt werden dafür auch praktische alle Datentypen, die mir gerade einfallen würden (number int/float, alle strings, boolean, enums -> multiple choice)
- aktuelle Änderungen betreffen vor allem die “Pflichtfelder” (bislang: id, location) die nicht mit Haecksen Clustern vereinbar sind und wenn wir ehrlich sind auch so auf der Karte

Probleme machen (Einträge ohne Koordinaten, problematische Namen in der Liste bei exotischen Ortsangaben wie “Frankfurt/Mainz/Wiesbaden”)
    - neue Pflichtfelder: id (autogenerated als neuer Eintrag wenn kleiner 0), type [space,group,individual,picnic], name (im Sinne eines Bezeichners), latitude und longitude (letztere für Haecksen individuals von County übernommen), email (als minimum requirement zur Kontaktaufnahme, fallback info@)
    - optional ist alles andere und alles andere kann theoretisch auch als privat markiert werden (außer boolean und arrays / multiple choice siehe Feature #25)
- außerdem gibts ne neue Tabelle für die County Daten, damit aus den gespeicherten Haecksen die Haecksen Cluster aggregiert werden können (Cluster werden nicht als Entitäten gespeichert)
- wenn irgendwann in der Zukunft flask oder fastapi den Zuschlag fürs Backend bekommen, dann kann der Code größtenteils recycelt werden

---

Revision #1
Created 2025-01-29 05:46:17 UTC by zombielypse
Updated 2025-01-29 05:46:49 UTC by zombielypse