**style.json metadataan muutoksia**

* trailmap:country
  + “all” - kaikissa versioissa
  + “fin” - vain Suomi
  + “global” - vain ei-maakohtainen versio (alussa siis ei-Suomi)
  + “nor”, “swe” jne. uusia maakohtaisia versioita
* trailmap:id\_legacy – alkuperäinen layer id

**Tiedostot**

* Alkupiste talteen, alkuperäiseen tyyliin EI kosketa: style.json —> style\_orig\_v2.json
  + Alkuperäinen style.json tarvitaan hyvin pitkään, vähintään n. puoleksi vuodeksi, tukemaan käyttäjien käytössä olevia mobiilisovelluksen versioita
* Uudelleen nimetyt “layers”: style\_orig\_v2.json —> style\_orig\_v2\_new\_ids.json
  + id\_export ja id\_import skriptien avulla
  + Nimeä skriptien jälkeen manuaalisesti: style\_orig\_v2\_new\_ids.json —> style\_base\_v2.json
* Perusversiot ja tasojen hallinta
  + style\_base\_v2.json (“täysi” style.json, ei muokata sellaisenaan missään)
  + style\_base\_v2\_export.xlsx (layers\_export skriptillä luotu edellisestä, “layers” hallintaan)
  + style\_base\_v2\_export.csv (Excel-taulukon tallennus csv-tiedostoksi)
  + style\_base\_v2\_import.json (layers\_import luo tämän, sen voi sitten manuaalisesti nimetä style\_base\_v2.json:ksi)
* Tyylin editointi tasojen sisällön yksityiskohtien osalta
  + style\_base\_v2\_edit.json (skriptillä konfiguroitu versio tyylin perusversiosta, valitut tasot näkyvissä)
  + style\_base\_v2\_edit\_final.json (ed. tiedoston versio, jossa muuttujat korvattu skriptillä)
  + style\_base\_v2\_old.json (alkuperäinen style\_base\_v2.json versio, layers\_commit-skripti tuottaa)
  + style\_base\_v2.json (muokkauksien jälkeen layers\_commit-skriptin tuottama uusi perusversio)
  + style\_v2.json (commit-skriptin tuottama tiilipalvelimelle menevä versio, tarvitaan vasta sitten kun muuttujat käytössä)

**Skriptit**

* + id\_export & id\_import - “layers” nimien vaihtoon, tarvitaan vain alussa
    - id\_export
      * INPUT: mikä tahansa style.json tiedosto
      * OUTPUT: <input file>\_ids\_exported.csv (taulukko, jossa riveinä “layers” style.json:sta)
    - id\_import
      * INPUT:
        + csv-tiedosto, jossa “layers” riveinä: 1. sarake alkuperäinen nimi, 2. sarake uusi nimi
        + style.json tiedosto, jolle muutos tehdään
      * OUTPUT: <input file>\_new\_ids.json
* v1\_migration
  + INPUT: style.json tiedosto joka alkuperäistä ”v1” formaattia
  + OUTPUT: <input>\_migrated.json – tekee tarvittavat muunnokset v2-tyyliä varten mm. source nimet, metadata formaatti jne.
  + layers\_export & layers\_import - “layers” hallintaan (järjestys, metadata)
    - layers\_export
      * Tasojen hallintaa varten “export”-toiminnolla luodaan tyylistä taulukko (xlsx), jossa tyylin tasot riveittäin ja sarakkeissa halutut konfiguraatio-elementit (export-skripti irrottaa tyylistä ne mitkä mahdollisia). Export voi myös hyödyntää edellistä taulukkoa (“merge”-operaatio), jotta saadaan taulukkoon tehdyt kommentit (jotka eivät style\_orig.csv:ssä) säilymään.
      * INPUT: mikä tahansa style.json tiedosto (ja käyttää myös oletuksena saman nimistä tiedostoa .csv päätteellä vanhaan metadataan, josta poimitaan kommentit mukaan)
      * OUTPUT: <input file>\_export.xlsx (taulukko, jossa riveinä “layers” ja jokaiselle niistä metadataa)
    - layers\_import
      * Taulukkoa muokataan halutusti ja muutosten jälkeen “import”-toiminnolla muunnetaan viimeisestä versiosta style\_base.json aina uusi pohjaversio style\_base.json.
      * Taulukon rivit määrittävät aina “import”-toiminnolla syntyvän style\_base.json:in tasot eli 1) Tasojen järjestys & metadata yms. ylemmät määrittelyt, 2) Poistetut tasot (jos ei ole taulukossa, niin ei tule style\_base.json:iin) ja 3) Uudet tasot (joissa aluksi vain nimi + metadata).
      * INPUT:
        + csv-tiedosto, joka luotiin layers\_export avulla (ja siihen tehdyt editoinnit toki ok) eli xlsx taulukko tallennettu csv:nä
        + style.json tiedosto, jolle muutos tehdään (yleensä style\_base\_v2.json)
      * OUTPUT:
        + <input>\_import.json
  + layers\_configure, apply\_variables, layers\_commit - “layers” tasojen yksityiskohtaiseen (renderoinnin) muokkaukseen
    - layers\_configure
      * Konfiguroi (configure) haluttu versio editointiin (“style\_edit.json”)
      * INPUT:
        + style.json tiedosto, jolle muutos tehdään (yleensä style\_base\_v2.json)
        + variant tag, joka määrää mikä tyylin variantti valitaan eli metadatan kentän trailmap:<variant tag> jonka arvo pitää olla “true”
        + country tag (optionaalinen), joka määrää mikä tyylin globaaliversio valitaan eli metadatan kentän trailmap:country arvo
        + hc tag (optionaalinen)
      * OUTPUT:
        + <input\_file>\_edit.json alkuperäinen style.json tiedosto, jossa “layers” konfiguroitu annetun syötteen perusteella (yleensä style\_base\_v2\_edit.json)
    - apply\_variables
      * watch-automatiikalla skripti korvaa siihen muuttujat ja tuottaa style.json tiedoston, jota Maputnik vahtii & näyttää livenä
      * TBD: watch-automatiikan toteutus per Mac vs. Windows
      * INPUT:
        + style.json tiedosto, johon muuttujat korvataan (yleensä style\_base\_v2\_edit.json)
        + muuttujat tiedosto, TBD formaatti
      * OUTPUT:
        + <input\_file>\_final.json (yleensä style\_base\_v2\_edit\_final.json)
    - layers\_commit

Editoinnit tehdään (esim. VSCode) joko layers\_configure tai apply\_variables tuottamaan style.json versioon, lopuksi tallennetaan commit-skriptin avulla tehdyt muutokset.

* + - * Commit-skripti ottaa sille annetusta style.json tiedostosta per layer “filter”, “layout” ja “paint” osiot ja yhdistää ne annettuun alkuperäiseen style.json tiedostoon (yleensä style\_base\_v2.json). Huom. jos tasoja on lisätty niin niitä ei huomioida, mutta niista annetaan varoitus. Tällä menettelyllä aiemmin luotu “layers”-taulukko (layers\_export skriptillä) on aina validi myös commit-skriptin jälkeen ts. skripti ei tallenna sellaisia muokkauksia, jotka olisivat päällekkäisiä taulukon kanssa (esim. tasojen uuden nimet, järjestys tms.).
      * Lisäksi skripti tuottaa normalisoidun style\_v2.json tiedoston, jossa on default-konfiguraatio (finland & mtb näin aluksi). Tämä tiedosto käytetään tiilipalvelimella.
      * INPUT:
        + Muokattu style.json tiedosto (yleensä joko style\_base\_v2\_edit.json tai muuttujien kanssa style\_base\_v2\_edit\_final.json)
        + Tyylin perusversio johon halutaan yhdistää muokkaukset (yleensä style\_base\_v2.json)
      * OUTPUT:
        + Alkuperäinen tyylin perusversio style.json tiedosto kopioidaan talteen <input file>\_old.json nimellä (eli yleensä style\_base\_v2\_old.json)
        + Uusi tyylin perusversio <input\_file2> (yleensä style\_base\_v2.json)
        + Uusi tiilipalvelimelle julkaistava versio style\_v2.json

**New style initial transformation**

* style.json —> style\_orig\_v2.json
  + Testaa onko eroja diff-komennolla, jos on niin uusi orig pitää luoda
* style\_ids.xlsx —> style\_ids.csv
* id\_import style\_ids.csv style\_orig\_v2.json , —> style\_orig\_v2\_new\_ids.json
  + Uudet tasonnimet
  + “touch” VSCodella ja save, jotta Prettier formatoi uudestaan
* mv style\_orig\_v2\_new\_ids.json style\_base\_v2.json
* v1\_migration style\_base\_v2.json —> style\_base\_v2\_migrated.json
  + New source names
  + Delete taustakartta source & layer that uses it
  + Transform layer metadata (high\_contrast)
  + Delete legacy mapbox metadata entries and top level id “bright”
* json\_validate style\_base\_v2\_migrated.json
* mv style\_base\_v2\_migrated.json style\_base\_v2.json