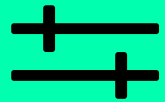


Mätsien (ProjectMatch) haku, filtteröinti ja algoritmi

1



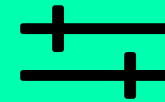
Oletuskriteerit
kantahauille kannassa
omassa taulussaan
(aluksi kovakoodattu,
myöhemmin myös UI)

2



Rust rikastaa dataa
algoritmin tuottamalla
osumaprosentilla mutta
ei tee muuta heavy
liftingiä

3



Fronttiin tehdään UI,
jolla tuloksia voi
edelleen filtteröidä ja
toki tyyllitellään
osumat

-Kun oletuskriteerit haulle haetaan kannasta, adminilla on kontrolli serverille aiheutuvasta kuormasta. Esim. haetaan kaikki kandidaatit joilla on yksikin mätsäävä taito vs. kolme

-Tällä tavalla serveri ei rasitu lopulta hirveästi. Frontissa ei enää tehdä uusia kantahakuja filtteröidessä vaan näytetään/piilotetaan olemassa olevan objektin osia. Tässä on toki mahdollisuus user errorille siten, että kriteerit kannassa ovat liian tiukat

-Algoritmin tuottama prosentti tehdään oletusarvojen pohjalta, sitä ei frontissa muuteta filtteröidessä

-Ensimmäinen ajatus algoritmista - CSS-specificityä myötäilevä pisteytys:

Mandatory = inline style

Availability = ID (tämä järjestys toimii hyvin vain jos oletuskriteerinä on min. 1 taito)

Muut mätsäävät skillit = class

Skillit mutta eivät riitä = element