Cum sa reduci cu 80% lungimea comenzilor din terminal

O problemă pe care o putem întâlni în dezvoltarea software este reprezentată de comenzile din terminal. Într-un un proiect ce respectă anumite standarde, precum folosirea de fișiere .env, Docker, Kubernetes sau testare, putem ajunge să avem un număr foarte mare de comenzi în terminal care, de cele mai multe ori, sunt și foarte lungi. Un exemplu concret ar fi dorința de a rula testarea atât în CI/CD, cât și local, poate chiar și în Docker. În acest caz, pentru aceeași testare vom scrie comanda, iar de fiecare dată va trebui să scriem în terminal export-urile pentru variabilele din environment (care, de cele mai multe ori, sunt destul de multe). Scrierea acestora în mod repetat de către developer scade viteza de dezvoltare, dar, cel mai important, scade atenția developerului de la task-urile principale.

O soluție pentru această problemă o reprezintă folosirea tool-urilor de creare de CLI commands, cunoscute și sub numele de „task runners”. Unul dintre cele mai populare tool-uri de acest tip este Makefile, însă posibilitățile sunt similare pentru toate aceste tool-uri. Pentru a exemplifica puterea acestora, am să mă raportez la experiența personală, mai precis la un tool numit „poethepoet” pentru Python, utilizat chiar în cadrul proiectului. Cum în procesul de dezvoltare aveam nevoie de diferite comenzi pentru Docker, testare, migrarea bazei de date sau curățarea cache-ului și a diferitelor fișiere generate de testare (toate acestea deseori depinzând de variabile din environment), am decis să folosesc acest tool. Pentru testarea end-to-end am folosit o bază de date nepersistentă pornita in Docker, iar testele au fost rulate local. În acest proces, aș fi fost nevoit să scriu în terminal 4 comenzi, având o dimensiune totală de 282 de caractere. Cu ajutorul tool-ului „poethepoet”, pot rula aceste 4 comenzi secvențial, folosind simpla comandă *poe test*.

Așadar, întreaga configurare de CLI commands poate că inițial va părea grea, dar, în opinia mea, pe termen lung, această opțiune facilitează dezvoltarea software într-un mod mai ușor și mai plăcut.

Referințe

Proiect MDS - https://github.com/hutanmihai/calorie-tracker-backend
Configurarea tool-ului in cadrul proiectului - https://github.com/hutanmihai/calorie-tracker-backend/blob/main/pyproject.toml
Poethepoet - https://github.com/nat-n/poethepoet
Makefile - https://opensource.com/article/18/8/what-how-makefile