

Letný report

Cieľ tohto semestra priamo nadväzoval na ciele minulého semestra.

Na základe dobrých vybudovaných základov na programovanie finálnej aplikácie, sme sa mohli pustiť priamo do implementácie.

Prvým krokom bolo spracovanie vstupu od užívateľa, implementované v súbore `parse-args.ts`.

Konkrétne sa jedná o funkciu `parseArgs`, ktorej vstupom sú pracovný adresár a argumenty od užívateľa.

Funkcia nastavuje prednastavené hodnoty, spracúva argumenty, nastavuje a upravuje premenné `outputPath`, `inputPath`.

Na tie sa v tejto funkcii pozerá iba ako na cesty v súborovom systéme, nie ako na reálne súbory.

Tieto cesty sa spracovaním stávajú absolútnymi cestami k potenciálnym súborom na disku.

Ďalšou dôležitou funkciou je `validateInputFile`, ktorá dostane `inputPath`, a jej úlohou je zistiť, či sa jedná o súbor, z ktorého vie užívateľ čítať.

Zisťovanie je realizované dvoma kontrolami, najprv kontrolou prístupových práv a následne kontrolou, či sa jedná o obyčajný súbor.

Nasleduje o čosi zložitejšia funkcia `validateOutputFile`, ktorá najprv testuje prístupové práva užívateľa, následne skúša vytvoriť súbor na ceste `outputPath`.

Taký pokus môže:

- vytvoriť súbor, čím sa program posúva ďalej v kóde
- zistiť že súbor existuje, a je potrebné potvrdenie od užívateľa o prepise obsahu súboru
- alebo sa súbor nedá otvoriť na zapisovanie, vtedy program končí.

Teraz poďme k súboru `package.json`.

Najpr niečo o rozhraní `IPkg`. To je celkom šikovné, keďže rovnaké rozhranie vieme použiť pre `package.json` od užívateľa, ale aj pre informácie o dependency moduloch.

Informácie o moduloch hľadáme na troch miestach a to v:

1. `package-lock.json`
2. `node_modules`
3. cez online register <https://registry.npmjs.org>.

Hľadaním na viacerých miestach znižujeme šancu, že sa nedostaneme k informáciám o module, ako napríklad licencia modulu a podobne.

Implementácia hľadania spočíva v identifikácii dependencies, ktorých informácia zatiaľ nemáme k dispozícii.

Teda každé spomenuté prehľadávané miesto prehľadáva postupne menšiu alebo rovnakú množinu modulov.

Ak sa informácie o module nepodarí nájsť vôbec, užívateľ je o tom informovaný cez štandardný výstup.

Ďalším, finálnym krokom príprav pred porovnávaním licencií je načítanie tabuľky licencií z csv súboru. Tento krok bol uľahčený použitím modulu csv-parse. Pomocou neho sme vytvorili pole, ktorého jeden riadok je informácia o konkrétnej licencií a hodnoty povolení alebo zákazov.

Posledným krokom je porovnávanie licencií.

Ak sú licencie rovnaké, nevznikajú žiadne konflikty licencií.

Ak však rovnaké nie sú, je potrebné sa pozrieť do tabuľky licencií a ich obmedzení a porovnať obmedzenia.

Toto porovnávanie je sprostredkované dvojrozmernou porovnávacou tabuľkou, kde index riadoku reprezentuje obmedzenie licencie závislosti a index stĺpca reprezentuje obmedzenie licencie modulu popísaného súborom package.json zo vstupu.

Ak porovnanie vráti !!!, potom ide o spor medzi licenciami.

Ináč, ak sa stĺpcové obmedzenie nerovná obmedzeniu z porovnania, je potrebné, aby užívateľ používaním svojho modulu neporušoval obmedzenie, ktoré je implikované výsledkom porovnania.