Adviesverslag

Project Lumber

Door Matthijs Booman

Inhoudsopgaven

Code structuur

Structuur van code is belangrijk in elke applicatie, het geeft overzicht om bepaalde code te vinden en eventueel te debuggen. Lumber is een project wat duidelijk een groot deel van het overzicht verloren heeft omdat het met verschillende ideeën in elkaar is gezet.

Om te beginnen is het duidelijk dat alles binnen index.ts onderdeel is van het hoofdproces, alle andere bestanden zijn vaak klein en spelen een kleine rol. Dit is geen probleem als er meerdere belangrijke dingen in zo'n bestand plaatsvinden, maar het zou goed zijn om alle kleine "utility" functies in 1 utils bestand te zetten.
Verder is het goed om te kijken of de code in de overige bestanden ook nog wel bruikbaar zijn, of het nog wel past bij het huidige idee en de vereisten van de applicatie.

De applicatie is geschreven in typescript, wat inhoudt dat er types & interfaces gemaakt en toegewezen zijn. In de huidige situatie zijn er een hoop interfaces die wel geëxporteerd worden uit de bestanden, maar nergens worden geïmporteerd. Om het overzichtelijk te houden zou ik controleren welke interface ook daadwerkelijk in andere bestanden gebruikt worden zodat alleen de interfaces die geëxport moeten worden dat ook zijn.
Het valt ook op dat er wat dubbele interfaces zijn die verspreid zijn over verschillende bestanden en dat sommige interfaces helemaal niet meer gebruikt worden. Ik raad aan om alle interfaces die in meerdere bestanden gebruikt worden te verzamelen in de al bestaande interfaces map.

Kwaliteit van de code

De applicatie is geschreven in typescript, dit houdt in dat het geschreven kan worden met de nieuwste ecmascript specificatie in gedachten omdat de typescript compiler de code om kan zetten naar elke versie van javascript.

Het is vrij duidelijk dat er geworsteld werd met nieuwe functies binnen javascript, zo wordt er vaak de map() functie van een array gebruikt als een for-loop. map() in plaats van een for-loop is in principe geen probleem, maar geeft wel een verkeerd beeld van wat dat deel van de code inhoud. Ik raad aan om alle code, vooral binnen de grotere classes, goed te analyseren en om bad practices te verbeteren.
Ook is er een hoop code die niet gebruikt wordt of geen toepassing (meer) heeft. Ik raad aan om stukken van de code te herschrijven zodat er beter gebruik gemaakt kan worden van nieuwe javascript functies en principes en zodat de niet gebruikte code geen onnodige code structuur achterlaat.

Error handling wordt wel toegepast, maar niet op duidelijke en uitgebreide wijze. Zo wordt er vaak alleen het error bericht gelogd zonder een stack trace, of worden er errors opgevangen en opnieuw aangemaakt waardoor de stack trace niet meer klopt. Ik raad aan om betere error handling te implementeren, al helemaal bij functies die bestanden aanmaken of aanpassen.

Veiligheid

De applicatie is een cli tool dat gemaakt is voor intern gebruik binnen het bedrijf, hierdoor hoef je geen rekening te houden met kwetsbaarheden van buitenaf zoals bijvoorbeeld veel webapplicaties hebben. Wel maakt de applicatie gebruik van node modules waarbij regelmatig kwetsbaarheden gevonden kunnen worden.

Om de applicatie zo veilig mogelijk te houden raad ik aan om altijd de applicatie bij te houden onder de nieuwste of lts vorm van nodejs. Zo zorg je dat er geen kwetsbaarheden van nodejs misbruikt kunnen worden.
Ook is het belangrijk om de dependencies (in dit geval de node modules) up to date te houden en regelmatig een audit uit te voeren. Een audit uitvoeren op node modules is vrij simpel, met de command “npm audit” wordt er direct gekeken of de geïnstalleerde versies van de dependencies kwetsbaarheden hebben.