

# Test plan

Project Lumber

Door Matthijs Booman

## Inhoudsopgave

### [Inhoudsopgave](#)

#### [Functioneel](#)

[Wat wordt er getest?](#)

[Wanneer wordt er getest?](#)

[Stappenplan](#)

#### [Systeem](#)

[Wat wordt er getest?](#)

[Wanneer wordt er getest?](#)

[Stappenplan](#)

#### [Continu / automatisch](#)

[Wat wordt er getest?](#)

[Wanneer wordt er getest?](#)

[Stappenplan](#)

#### [Acceptatie](#)

[Wat wordt er getest?](#)

[Wanneer wordt er getest?](#)

[Stappenplan](#)

# Functioneel

## Wat wordt er getest?

De vereisten van een toegevoegde / aangepaste functie.

## Wanneer wordt er getest?

De functie wordt getest zodra de developer denk dat alle vereisten van die functie geïmplementeerd of verbeterd zijn.

## Stappenplan

1. Controleer of de geschreven code van de functie voldoet aan de beschreven functionaliteit van de taak.
  - a. Zodra de functie niet voldoet aan de beschreven functionaliteit moet er gekeken worden wat er binnen de functionaliteit mist en worden geïmplementeerd.
2. Voer de functie uit op een (voorbeeld) use case project waarbij het duidelijk is of de functie veranderingen maakt.
3. Controleer of het resultaat van de functie voldoet aan de verwachting van de taak.
  - a. Zodra er errors plaatsvinden of de functie heeft zich niet voldaan aan de verwachtingen dient er gekeken te worden naar wat er fout ging en dat eerst rechtgetrokken te worden voordat er opnieuw getest wordt.

# Systeem

## Wat wordt er getest?

Het hele programma.

## Wanneer wordt er getest?

Lumber wordt getest zodra de functionele testen succesvol voltooid zijn.

## Stappenplan

1. Pak een (voorbeeld) use case project dat in originele staat is.
2. Geef in de config.json van Lumber aan waar de root module zit.
3. Voor Lumber uit met de command "npm start" of "npx ts-node index"
4. Controleer of Lumber alle benodigde bestanden heeft gegenereerd.
5. Open de browser en controleer of de (voorbeeld) use case werkt volgens zoals verwacht.
  - a. Zodra Lumber een bestand niet (goed) gegenereerd heeft of de (voorbeeld) use case werkt niet zoals verwacht dienen de fouten te worden opgezocht en de functionele testen opnieuw te worden uitgevoerd.

# Continu / automatisch

## Wat wordt er getest?

Het hele programma wordt getest in een geïsoleerde en geautomatiseerde omgeving.

## Wanneer wordt er getest?

Zodra systeem testing voltooid is en alle code is gecommitt naar de issue branch.

De geautomatiseerde tests worden getriggerd zodra de code van de issue binnen gitlab is gemerged met de w-nummer branch van de taak.

## Stappenplan

1. Zorg dat alle code is gecommitt naar de issue branch.
2. Merge de issue branch met de w-nummer branch van de huidige taak.
3. Kijk naar de response van de CI van gitlab.

# Acceptatie

## Wat wordt er getest?

Of het programma klaar is voor een nieuwe uitrol.

## Wanneer wordt er getest?

Wanneer er nieuwe functionaliteiten zijn toegevoegd of wanneer bugs / overige fouten zijn opgelost.

## Stappenplan

1. Is het een update met nieuwe functionaliteiten of lost het bugs en overige fouten op?
  - a. Nieuwe functionaliteiten: bespreek welke nieuwe functionaliteiten geïmplementeerd moeten zijn.
  - b. Bugfix: Zodra de bugs / overige fouten opgelost zijn.
2. Controleer of alle automatische tests op de w-nummer branches voltooid zijn zonder problemen.
3. Merge de w-nummer branches naar een uniek aangewezen u-nummer branch.
4. Voor een systeem / automatische test uit op de u-nummer branch versie van het programma.
  - a. Zodra de test geslaagd is dient er een uitrol getriggerd te worden waarbij gitlab een nieuwe tag geeft aan deze versie van de code en het als npm package gepubliceerd wordt.