

Requirements - Co-reflection

Introductie

In dit onderzoeksdocument vind je de resultaten en conclusie over wat de requirements zijn voor het ontwikkelen en beheren van de front-end componenten library.

Requirements - Co-reflection	1
Introductie	1
Doel	2
Motivatie	2
Hoofdvraag	2
Hypothese	2
Deelvragen	2
Plan van aanpak	2
Resultaten	3
Requirements	3
Beveiliging	3
Testing (guidelines)	3
Packages	3
Documentatie	3
Overig	3

Doel

Het hoofddoel is om vast te stellen wat de requirements zijn om de front-end componenten library succesvol te ontwikkelen en te beheren.

Motivatie

Als er geen requirements vastgesteld worden, zal de front-end componenten library hoogstwaarschijnlijk niet goed functioneren en/of niet geschikt zijn voor BBD.

Hoofdvraag

Wat zijn de requirements voor het ontwikkelen en onderhouden van de front-end componenten library?

Hypothese

Er wordt verwacht dat Bluebird Day zelf een aantal specifieke technische eisen heeft, bijv. op het gebied van beveiliging/account management. Verder wordt er verwacht dat er eisen zijn op het gebied van content; wat moet er in de front-end componenten library staan en wat niet?

Deelvragen

- Wat zijn de eisen vanuit Bluebird Day omtrent de beveiliging van het platform?
- Wat zijn de eisen vanuit Bluebird Day omtrent de code kwaliteit/coding standaarden?
- Wat zijn de eisen die na de conclusies uit het onderzoek nodig zijn?
- Welke informatie heeft de BBD front-end developers nodig om de library te kunnen gebruiken?

Plan van aanpak

Showroom research wordt toegepast.

De 'Co-reflection' methode wordt ingezet. Samen met de front-end lead developer en bedrijfsbegeleider wordt er gediscussieerd over waaraan het platform moet voldoen.

Resources die gebruikt zullen worden om requirements op te stellen zijn het BBD beleid (bijv. beveiliging en coding standaard regels) en de conclusies uit de eerdere onderzoeksresultaten.

Resultaten

Requirements

Beveiliging

- Toegang - wie heeft toegang nodig?
 - Iedereen binnen <https://git.bluebirdday.nl/>
- GitLab tokens - hoe worden tokens op een veilige manier gebruikt?
 - **CI_JOB_TOKEN**
 - **Personal access token**
 - **Group deploy token**
 - **Job token**

Testing (guidelines)

- GitLab CI - welke CI tests zijn er nodig om errors in de remote repo te voorkomen?
- Code kwaliteit - hoe wordt de kwaliteit van de componenten bewaakt?
 - **Code reviews (dezelfde flow aanhouden zoals tijdens projecten gedaan wordt?)**

Packages

- Package Registry settings - welke settings zijn nodig?
 - **Number of duplicate assets to keep: 1**
 - **Settings > Usage Quotas > Storage voor opslag van de packages**
- Package command - welke package manager commands worden gebruikt?
 - **Yarn; de Bluebird Day standaard**
 - **NPM; tweede keus**
- Package Managers - welke package managers zijn er nodig?
 - **NPM**
 - **Composer**

Documentatie/Guidelines/Regels

- Onderhoud - hoe wordt documentatie onderhouden?
 - **Documentatie wordt na een commit automatisch bijgewerkt.**
- Content - welke content is er nodig?
 - **Authenticeren/linken naar de package registry**
 - **Packages installeren in een nieuw project**
 - **Packages importeren**
 - **Coding standaard voor het maken van herbruikbare components**
 - **Nieuwe packages registreren in/uploaden naar de package registry**
 - **Documentatie schrijven voor een toegevoegde component; API usage**

Regels

Hoe zorgen we ervoor dat de library netjes bijgehouden wordt?

In de library zullen er verschillende type componenten aanwezig zijn. Deze types hebben verschillende prioriteiten.

BBD werkt met Atomic design.

Atoms zijn elementen zoals een form label, input field & button. Elementen die gegroepeerd moeten staan worden Molecules genoemd. Meerdere Molecules bij elkaar worden Organisms genoemd, bijvoorbeeld een header. Meerdere organisms en molecules bij elkaar vormen een template. Dit is vaak één webpagina. Pages zijn instances van templates, de placeholder content is dan vervangen met representatieve content die de gebruiker moet zien. Denk aan afbeeldingen.

Must have

Molecules; molecules zoals checkbox fields, buttons, breadcrumbs, cards en pagination worden veel toegepast door BBD. Molecules zijn vaak goed te hergebruiken voor elk project, daarom bieden we veel Docs pages, zoals die van Vue StoreFront, molecules componenten aan.

Should have

Organisms; headers en footers zijn veelgebruikte organisms, deze sections kun je zien als een mini template. Headers bevatten namelijk vaak dezelfde soort molecules, maar per project kunnen deze molecules in een andere volgorde nodig zijn. Het gebruiken van organisms kan de development tijd behoorlijk versnellen.

Could have

Atoms; voor elk project heeft BBD wel een styleguide met daarin de benodigde kleuren, fonts en andere atomics. Dit is echter project gericht, en daarom niet zo handig om als aparte componenten aan te bieden.

Voor de BBD componenten library is dit daarom niet van het grootste belang, maar wel handig om op dezelfde plek te hebben aangezien de developers de unieke kleuren en fonts nodig hebben om molecules mee te bouwen. In de huidige situatie staan de style guides op een aparte plek voor elk project.

- Simpele (transition) animaties; elementen die inladen wanneer de gebruiker scrolled, kunnen een eenvoudige transition property bevatten om een simpele animatie te tonen. Hover animations voor buttons is nog een voorbeeld.

Won't have

Templates & Pages; BBD's klanten eisen websites die bij het bedrijf passen en uniek zijn, templates en pages zijn te specifiek en worden dus niet gebruikt.

Geavanceerde animaties; deze animaties zijn moeilijk modulair te maken en kunnen mogelijk alleen worden aangeboden als voorbeeld.

Conclusie

Nu het onderzoek klaar is, kan de hoofdvraag worden beantwoordt.

“Wat zijn de requirements voor het ontwikkelen en onderhouden van de front-end componenten library?”

De requirements zijn het hebben van een privé/beveiligde omgeving/systeem en het hebben van test en documentatie guidelines om de library te kunnen onderhouden met daarnaast ruimte voor uitbreiding/support mochten er andere package types nodig zijn.