

Advies

Frontend Componenten Library



Bluebird Day.

a Handpicked agency

Inleiding	3
Doel	3
Motivatie	3
Conclusies	4
Vervolgstappen	4

Inleiding

In dit document wordt een advies gegeven aan Bluebird Day wat betreft het ontwikkelen en inzetten van een front-end componenten library voor het bedrijf Bluebird Day.

Doel

Het hoofddoel is om een advies te geven dat ervoor zorgt dat Bluebird Day een front-end componenten library kan ontwikkelen en onderhouden dat aansluit op de requirements.

Motivatie

Met dit advies kan Bluebird Day zelfstandig aan de slag zonder mij als 'front-end componenten library' expert nodig te hebben, mocht ik er niet meer zijn na afloop van mijn stageperiode.

Hoofdvraag

Wat voor advies moet er gegeven worden om ervoor te zorgen dat Bluebird Day zelfstandig een front-end componenten library kan ontwikkelen en onderhouden?

Conclusies

Na afloop van mijn afstudeeropdracht en vele onderzoeken, kan het volgende geconcludeerd en geadviseerd worden:

- De [GitLab Package Registry](#) feature is het meest geschikte middel om front-end componenten op te slaan en op te kunnen halen. Er wordt dus geadviseerd de Gitlab Package Registry te gebruiken. Ook voor andere soorten componenten (bijv. back-end componenten) wordt geadviseerd de Gitlab Package Registry te gebruiken.
- [GitBook](#) is de meest geschikte tool om documentatie voor de componenten te maken en te onderhouden. Er wordt dus geadviseerd GitBook te gebruiken voor de technische documentatie.
- [Replit](#) is de meest geschikte tool om code van componenten te tonen/'previewen' in de documentatie van componenten.

Vervolgstappen

1. **Confluence en/of GitBook;** maak een afweging of het mogelijk is van Confluence over te stappen naar GitBook vanuit een financieel perspectief en vanuit een organisatieperspectief. Het advies is om de huidige documentatie op Confluence te exporteren naar GitBook. GitBook wordt dan gebruikt voor technische en niet-technische documentatie binnen Bluebird Day. Een mogelijk probleem is dat GitBook duurder is dan Confluence en het migreren naar GitBook qua besteedde tijd het niet waard is.
2. **Testen design systeem;** de structuur van de library moet voor developers duidelijk zijn en uitbreidbaar zijn. Uit onderzoek bleek dat veel developers van Bluebird Day bekend zijn met 'Atomic design'. Daarom wordt aangeraden het Atomic design systeem te gebruiken in de structuur van de library. Ook moet er rekening gehouden worden met de verschillende type componenten (Magento, Shopware, Vue Storefront). Er wordt aangeraden te testen hoe de library het beste gestructureerd kan worden.
3. **Onderzoek dependency management;** tijdens de co-creation/reflection meeting met de developers, bleek dat het managen van de dependencies niet goed verloopt. Nested dependencies worden niet goed behandeld. Met [NPM workspaces](#) zou dit opgelost kunnen worden. Ook het toevoegen van een [CHANGELOG.md](#) kan helpen het versiebeheer overzichtelijk te houden. Hier moet nog uitgebreid onderzoek naar gedaan worden.
4. **Motiveer de developers;** zorg voor een basisset van componenten. De meest gebruikte componenten in de projecten moeten in de componenten library komen te staan, zodat de developers meer motivatie hebben om het te gebruiken. Zodra developers het gaan gebruiken en merken dat het handig is, zullen zij ook meer motivatie hebben tijd te besteden aan het toevoegen van nieuwe componenten. Zo krijgt de componenten library steeds meer waarde voor de developers en zullen zij de library vaker gaan gebruiken.