

NPM Literatuuronderzoek

Introductie

In de voorafgaande trendanalyse werd geconcludeerd dat NPM mogelijk de functionaliteiten aanbiedt die BBD nodig heeft. In dit onderzoek wordt er een 'deep-dive' gedaan om de NPM functionaliteiten beter te begrijpen. In dit onderzoeksdocument vind je de resultaten en conclusie van het literatuuronderzoek naar NPM (Node Package Manager).

Introductie	1
Doel	2
Motivatie	2
Hoofdvraag	2
Hypothese	2
Deelvragen	2
Plan van aanpak	2
Resultaten	3
Hoe worden packages geupload naar NPM?	3
Benodigdheden	3
Uploaden openbare packages	3
Uploaden privé packages	3
Hoe worden packages onderhouden? Is er een digitale omgeving (bijv. een dashboard) voor het managen van de packages?	4
NPM organization	4
GitLab NPM Registry	5
Conclusie & Aanbevelingen	7
Bronnen	7

Doel

Het hoofddoel is om de opties om met NPM packages/componenten te beheren, te leren kennen.

Motivatie

Als de functionaliteiten van NPM bekend zijn, kan er beslist worden of NPM voldoet aan de eisen/wensen/behoefte van BBD.

Hoofdvraag

Hoe werken de functionaliteiten die NPM aanbiedt om packages te kunnen beheren?

Hypothese

Er wordt verwacht dat NPM een optie heeft om packages privé te uploaden en een soort omgeving om de packages te beheren.

Deelvragen

- Hoe worden packages geupload naar NPM?
- Hoe worden packages onderhouden? Is er een digitale omgeving (bijv. een dashboard) voor het managen van de packages?

Plan van aanpak

Desktop en workshop research wordt toegepast.

NPM wordt getest door een package privé te uploaden naar NPM. Om dit uit te voeren moet er eerst kennis opgedaan worden door de NPM docs te bekijken.

Resultaten

Hoe worden packages geupload naar NPM?

NPM packages kunnen openbaar of privé zijn. Hoe ze precies worden geupload, verschilt. (Creating and publishing scoped public packages | npm Docs, z.d.)

Benodigdheden

- NPM
 - Git
-
1. Log in met je NPM account.
 2. Maak een directory aan voor je package
 3. Navigeer naar je package
 4. Link een git repository
 5. Gebruik '**npm init --scope=@my-username**' of '**npm init --scope=@my-org**' om de scope door te geven aan NPM. Op 'my-org' hoort je NPM organisatienaam te staan als je een organized-scoped package wilt aanmaken.

Uploaden openbare packages

Gebruik '**npm publish --access public**'

Uploaden privé packages

Gebruik '**npm publish**'

Packages kunnen na het uploaden alsnog omgezet worden van 'public' naar 'private' en andersom:


Status	Command
Van 'public' naar 'private'	' npm access restricted <package-name> '
Van 'private' naar 'public'	' npm access public <package-name> '

Van 'public' naar 'private' -> '**npm access restricted <package-name>**'
'**npm access public <package-name>**'



Hoe worden packages onderhouden? Is er een digitale omgeving (bijv. een dashboard) voor het managen van de packages?


NPM organization

NPM geeft je de mogelijkheid een 'organization' aanmaken. Hiermee heb je een eigen omgeving waarin je je packages kunt managen. Je kunt zoveel 'public' packages uploaden als je wilt, maar voor private packages is er betaling nodig (\$7 per maand). *npm. (n.d.)*

 next() Packaged Middleware


ProductsPricingDocumentation


 



bbd-testing

Rick van Wijk

 0 Packages

 1 Organization

Organization 1

rickvanwijk

GitLab NPM Registry

Een manier om zonder premium plan private packages te kunnen beheren, is door gebruik te maken van de GitLab NPM Registry. GitHub heeft overigens een soortgelijke feature, maar aangezien BBD gebruikmaakt van GitLab, wordt er gekozen om GitLab's NPM Registry te testen.


Gebruikmaken van de privé omgeving

1. Authenticate met de package registry
 - Met een personal access token
 - Met een deploy token
 - Met een OAuth token
 - Met een CI job token
2. Maak een GitLab repo aan
3. Voeg een .npmrc file toe en voeg het 'npm endpoint' toe
4. Publish je package met '**npm publish**'

Er zijn [meerdere manieren mogelijk en stappen](#) nodig om een package te uploaden naar een privé omgeving (GitLab 2022 - NPM Packages in the package registry).

Rick van Wijk > Bbd Frontend > Package Registry

Package Registry

 2 Packages


Filter results



Q

Version ▾

↓

@rvanwijk/bbd-card latest


0.1.0 published by Rick van Wijk  npm



 main -> [2dcef138](#) 

Created 21 hours ago

⋮

@rvanwijk/bbd-frontend

0.1.0 published by Rick van Wijk  npm

 main -> [aa7ebfec](#) 

Created 6 days ago





⋮

@rvanwijk/bbd-card

Delete

v 0.1.0 published 1 day ago [latest](#) npm  184.88 KiB  main[Detail](#) [Other versions](#)

History

-  @rvanwijk/bbd-card version 0.1.0 was first created 1 day ago
-  Created by commit [#2dcef138](#) on branch **main**
-  Built by pipeline [#67959](#) triggered 1 day ago by Rick van Wijk
-  Published to the **Bbd Frontend** Package Registry 1 day ago

Installation

Show NPM commands ▾

```
npm i @rvanwijk/bbd-card
```

Registry setup


☒ Instance-level☐ Project-level

```
echo @rvanwijk:registry=https://git.bluebirdday.nl/api/v4/packages/npm/ >> .npmrc
```

You may also need to setup authentication using an auth token. [See the documentation](#) to find out more.

Assets

Delete selected

<input type="checkbox"/>	Name	Size	Created
<input type="checkbox"/>	▼  @rvanwijk/bbd-card-0.1.0.tgz	184.88 KiB	1 day ago ⋮

Support

Zoals bekend werd tijdens de interviews, gebruikt BBD veel verschillende programmeertalen en frameworks. NPM is alleen geschikt voor JavaScript/TypeScript packages dus kan NPM ingezet worden voor projecten waar bijv. Vue Storefront wordt gebruikt, maar BBD maakt ook gebruik van PHP. GitLab Package Registry biedt support voor Composer, wat in principe gezien kan worden als de NPM voor PHP packages.

Package type	GitLab version
Composer	13.2+
Conan	12.6+
Go	13.1+
Maven	11.3+
npm	11.7+
NuGet	12.8+
PyPI	12.10+
Generic packages	13.5+
Helm Charts	14.1+

Conclusie & Aanbevelingen

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

“Hoe werken de functionaliteiten die NPM aanbiedt om packages te kunnen beheren?”

In combinatie met GitLab biedt NPM een omgeving aan waarin packages beheerd kunnen worden d.m.v van Git version control. Het uploaden van packages wordt gedaan door te authenticeren met de package registry van GitLab, een npm endpoint te zetten in je GitLab repo en deze te publishen met ‘npm publish’.

Het beheren van packages wordt grotendeels door GitLab gedaan. Features zoals het opzoeken/filteren van packages en automatische documentatie zijn allemaal te vinden in de GitLab repo. Wel ben je er als developer verantwoordelijk voor dat je repo structuur op orde blijft en dat zelfgemaakte packages geoptimaliseerd zijn (door de packages te ‘minifyen’ met [rollup.js](#) bijvoorbeeld).

Hoe nu verder?

Support voor PHP moet getest worden aangezien Magento PHP gebruikt en Magento in de toekomst nog regelmatig gebruikt zal worden door BBD.

Verder is het verstandig een repo structuur te creëren wat ervoor moet zorgen dat BBD front-end developers makkelijk packages kunnen uploaden en vinden. Denk hierbij aan package benamingen en folder structuur.

Bronnen

- npm. (n.d.). Retrieved September 26, 2022, from <https://www.npmjs.com/>
- Creating and publishing scoped public packages | npm Docs. (z.d.). Geraadpleegd op 11 oktober 2022, van <https://docs.npmjs.com/creating-and-publishing-scoped-public-packages/>
- NPM Packages in the package registry. GitLab. (n.d.). Retrieved October 11, 2022, from https://docs.gitlab.com/ee/user/packages/npm_registry/