Onderzoek CMS

Created	@September 30, 2024 10:27 AM
Tags	

Welk CMS kan ik het beste gebruiken voor mijn portfolio-website?

Deelvragen:

Headless CMS

- [x] ~~Wanneer gebruik ik een headless CMS, en wanneer juist een traditioneel CMS?~~
- [x] ~~Wat zijn de limitaties van een headless CMS?~~
- [x] ~~Hoe veilig is een headless CMS?~~

Hosting

- [x] ~~Hoe/waar worden CMS'en gerund?~~

Voor module 5.2 van de opleiding Web Developer aan het Grafisch Lyceum Utrecht moesten wij een aantal contentmanagementsystemen onderzoeken. Naar aanleiding van dit onderzoek moesten we één CMS uitkiezen, waarin we een project gingen bouwen. Als project bouw ik een Portfoliowebsite.

Voor dit onderzoek ga ik CMS'en in het algemeen en 2 verschillende systemen onderzoeken, deze systemen zijn:

- Drupal (https://www.drupal.org/home)
- Strapi (https://strapi.io/)

Eventuele bronnen die behulpzaam kunnen zijn:

- https://stackshare.io/stackups/trending (site met vergelijkingen van tech (stacks))
- https://www.drupal.org/docs/develop/decoupled-drupal (documentatie voor headless Drupal)
- https://www.reddit.com/r/node/ (Subreddit voor Node.js, goed voor gebruikerservaringen)
- https://vercel.com/guides/using-a-headless-cms-with-vercel (overzicht van CMS voorbeelden met Next.js)
- https://hygraph.com/blog/cms-security (CMS security)
- https://www.drupal.org/docs/develop/local-server-setup (Lokale Drupal setup)

CMS (in het algemeen)

CMS staat voor contentmanagementsysteem (ja, het is echt maar 1 woord). Met dit systeem kunnen mensen met een login de inhoud van een site aanpassen, zonder aan de code te zitten. Dit is handig omdat veel sites die worden gemaakt voor een klant, ook beheerd moeten worden. Het is wel mogelijk om elke keer handmatig de code aan te passen, maar het is natuurlijk makkelijker voor de klant en voor de developer als de klant dit zelf kan.

Onder de term 'Content Management System' valt heel veel, daarom zijn er ook zoveel verschillende opties, elke optie heeft zijn sterke punten.

Verschillende opties:

Headful/traditioneel CMS: Een traditioneel CMS regelt zowel de backend als de frontend. Het CMS is deel van de website.

Headless CMS: Een headless CMS is in tegenstelling tot een headful CMS losgekoppeld van de frontend. Een headless CMS levert dus alleen de data aan de website en regelt niet de frontend.

Open source CMS: een open source CMS draait meestal lokaal en is in grote mate aanpasbaar. Meestal heeft een open source CMS veel meer setup nodig dan een closed source CMS.

Closed source CMS: Een closed source CMS draait meestal in de cloud en heeft nauwelijks setup nodig. Denk aan websitebouwers zoals Wix.

Wanneer kies ik voor een headless CMS?

Een headless CMS wordt vaak gebruikt bij websites die meer maatwerk nodig hebben als het gaat om frontend. Veel geavanceerde CSS- of JavaScript functionaliteiten zijn heel moeilijk te doen wanneer een gebruik wordt gemaakt van een headful CMS. Ook kan een Headless CMS gebruikt worden om data te leveren aan verschillende bronnen, zoals een website en een mobiele app, die dezelfde data nodig hebben. Omdat een headless CMS niet direct aan de frontend is gekoppeld, is het moeilijker om binnen te breken en significante aanpassingen aan de site te doen. De API endpoints worden ook vaak beveiligd met RBAC.

Wanneer kies ik juist voor een traditioneel CMS?

Een traditioneel CMS kan veel handiger zijn als een klant niet alleen data, maar ook het uiterlijk van de website aan wilt passen. Vaak maakt een website met een headful CMS gebruik van een WYSIWYG editor, wat ervoor zorgt dat de klant echt alles aan de website kunnen aanpassen.

Hoe worden de systemen gehost?

De meeste open source systemen worden lokaal gehost. Voor CMS’en die op PHP draaien wordt het over het algemeen aangeraden om een Docker-

gebaseerde oplossing te gebruiken, maar met een programma als XAMPP is het natuurlijk ook mogelijk.

Er zijn ook een aantal Node.js gebaseerde systemen (denk aan Strapi, Contentful of Gatsby) die vaak werken met de NPM CLI. Om ze te builden, run je vaak `npm run build`. Om vervolgens het project te starten, run je meestal `npm start`.

Verder heb je ook nog de closed-source contentmanagementsystemen zoals Contentful en Squarespace, die in de cloud worden gehost.

CMS 1: Drupal

Wat is Drupal?

Drupal is een open source CMS gebouwd in PHP. Hoewel het systeem in 2001 is gereleased, wordt het vandaag de dag nog steeds gebruikt door veel bedrijven, onder deze bedrijven vallen zo ook Tesla, NASA, Nokia en UNICEF (bron: https://smartbees.co/blog/who-uses-drupal-20-famous-drupal-websites). De voornaamste reden dat deze zeer grote/professionele bedrijven Drupal gebruiken is om de veiligheid; Drupal staat bekend als een van de meest veilige open source CMS'en. Drupal biedt zelf een aantal core-functionaliteiten aan, maar er bestaan ook modules voor, om de functionaliteit nog verder uit te kunnen breiden. Op dit moment zijn er ruim 52.000 modules, allemaal te vinden op https://www.drupal.org/project/project_module. Het systeem kan zowel traditioneel als headless worden gebruikt, waardoor het voor veel verschillende doeleinden gebruikt kan worden.

Bevindingen

Om überhaupt Drupal op te zetten was ik meer dan anderhalf uur kwijt. De stappen die ik moest ondernemen om errors op te lossen:

- De `gd` extension in `php.ini` aanzetten
- Composer updaten
- PHP volledig opnieuw installeren, omdat de versies niet compatible waren
- De PDO extension in `php.ini` aanzetten

- Opcache in `php.ini` aanzetten, hier was zeker geen goede documentatie voor

Nadat ik al deze errors had opgelost en de basic setup had gedaan kreeg ik deze error in mijn `error.log`:

```
Uncaught PHP Exception Twig\\Error\\LoaderError: "Template "C
```

Hierna kwam ik tot de conclusie dat Drupal voor mij niet de beste keuze was voor mijn relatief simpele project, omdat alleen het opzetten al zo veel tijd kostte.

CMS 2: Strapi

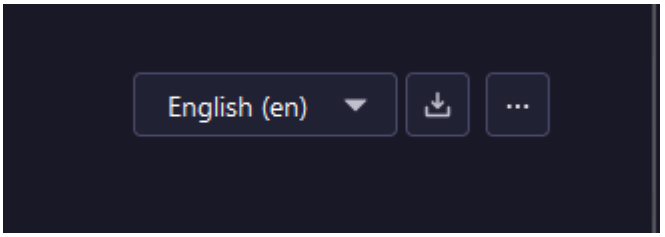
Wat is Strapi?

Strapi is ook een open source CMS, maar runt in tegenstelling tot Drupal op Node.js. Hoewel je met Drupal ook de front-end kan maken, is Strapi volledig headless en is daarom ook gebouwd met een API-first approach. Om data te fetchen van de Strapi API zijn er 2 opties: RESTful en GraphQL. Het grootste verschil tussen die twee is het feit dat REST API's vaak meerdere endpoints hebben, zoals `/api/user/9870769697` en `/api/users`. GraphQL werkt daarin tegen meestal met maar 1 endpoint, zoals `/graphql`. Je kan dan queries uitvoeren door queries zoals deze te sturen:

```
query Tests {
  tests {
    description
    title
    created_at
  }
}
```

Doordat veel queries bijna volledig dynamisch zijn, is het een stuk minder veilig dan RESTful API's. Hier maak je dus een keuze tussen veiligheid (REST) en flexibiliteit (GraphQL) (bron en meer informatie: https://strapi.io/blog/graph-ql-vs-rest-how-to-make-the-right-choice).

Bij de installatie van Strapi zitten ook al hele handige features, zoals internationalization. Hierdoor kan je voor elke entry verschillende versies maken, voor verschillende talen. Voor mijn portfolio wil ik 2 talen toevoegen: Engels en Nederlands. Een package genaamd next-intl (https://www.npmjs.com/package/next-intl) zorgt voor de routing en levert hiermee de locale (zoals 'nl' of 'en'). Vervolgens kan je een query sturen naar het cms met een 'locale' parameter, hierdoor krijg je alleen de data in een bepaalde taal. Meer informatie over de implementatie van internationalization kan makkelijk gevonden worden op de site van Strapi: https://strapi.io/features/internationalization.



Bevindingen

Het installeren van Strapi ging heel makkelijk: binnen 15 minuten had ik het volledige cms werkend. Je runt een NPX command (`npx create-strapi-app@latest my-project`), vult een paar opties in (zoals TypeScript, database credentials) en het project wordt voor jou gebouwd. Nadat alles geïnstalleerd is (duurde bij mij ongeveer 10 minuten) run je deze commands om in het admin panel te komen:

```
cd jouw-project-dir
npm run develop
```

Om de database vervolgens te querien, gebruik je een tool als Axios. Zo'n query kan er als volgt uit zien:

```
const response = (
    await axios({
        url: apiEndpoint,
        method: "POST",
        headers: {
            "Content-Type": "application/json",
            Accept: "application/json",
            Authorization: `Bearer ${process.env.STRAPI_TOKEN
```

```
        },
        data: {
            query: `
                query($locale: I18NLocaleCode) {
                    homepage(locale: $locale) {
                        locationHeader
                        startedProgrammingHeader
                        startedDesigningHeader
                        favAlbumHeader
                        introText
                        techStackHeader
                        aboutMeTexts {
                            text
                            image {
                                url
                            }
                        }
                    }
                    global {
                        dateOfBirth
                        designingSince
                        programmingSince
                        city
                        country
                    }
                    skills(pagination: { start: 0, limit: 100
                        name
                        icon {
                            url
                        }
                        inTechStack
                        confidenceLevel
                    }
                }
            `,
            variables: { locale },
        },
```

```
    })
).data;
```

Vervolgens manipuleer ik de data met een paar classes en pass ik die data naar de Client Components van Next.js.

Vergeleken met Drupal vind ik Strapi veel makkelijker te gebruiken.

Conclusie

De afgelopen week heb ik 2 verschillende contentmanagementsystemen uitgeprobeerd. De criteria waarop ik mijn conclusie heb gebaseerd:

- Gebruiksvriendelijkheid voor een klant
- Tijd om het CMS werkend te krijgen
- Integratie met een bestaand project

Gebaseerd op deze criteria, denk ik dat Strapi het beste bij mijn project past. Van alle systemen die ik had geprobeerd was het verreweg het meest intuïtief, zowel het opzetten als het invoeren van de data. Voor mijn portfolio gebruik ik dus Strapi.