

Vergelijkende Analyse van Content Management Systemen

HET IDEALE CMS IDENTIFICEREN VOOR
'VESTIZ'

LUKAS OLIVIER, NIELS SOETE EN LUCAS GUILLEMYN

Contents

Introductie.....	4
Onderzoek.....	6
Sanity.....	6
Installatie.....	6
Configuratie.....	7
Thema's/ Templates.....	8
Uitbreidingen.....	8
Bouwen en gebruik maken van RESTful API's.....	9
Meertalige ondersteuning.....	10
Versiebeheer met GIT en de implementatie van CI/CD-Pipelines.....	11
Implementatie en onderhoud.....	12
Beveiliging.....	13
Conclusie.....	14
1. Installatie.....	15
Stappen voor de Lokale Installatie van Ghost CMS.....	16
2. Configuratiemogelijkheden.....	17
Thema's en Ontwerpaanpassingen.....	17
Mail.....	17
Publicatie-instellingen.....	17
Gebruikersbeheer.....	17
Integraties.....	17
Lidmaatschappen en Betalingen.....	17
Meertalige Ondersteuning.....	17
2. Thema's/ templates.....	18
Thema's kiezen en aanpassen.....	18
Templates bewerken.....	18
3. Uitbreidingen.....	19
Apps en integraties installeren.....	19
Aangepaste code toevoegen.....	19
4. Bouwen en gebruik maken van RESTful API's.....	20
Content en Admin API's.....	20
Gebruiksvriendelijke Libraries.....	20
Eenvoudige Beheerdersauthenticatie.....	20
Uitgebreide Documentatie.....	20
Praktische voorbeelden.....	21
5. Meertalige ondersteuning.....	22
6. Versiebeheer met GIT.....	23
7. Security.....	23
Automatische SSL.....	24
Standaardtoestemmingen.....	24

Brute force protection.....	24
Data validatie and serialisatie.....	24
Encoded tokens.....	24
Wachtwoord hashing.....	24
SQLi preventie.....	24
XSS preventie.....	24
1. Installatie.....	26
Stappen voor het maken van een project met Strapi.....	26
2. Configuratiemogelijkheden.....	27
Content Types.....	27
API-Configuratie.....	27
Plugin-Extenties.....	28
Gebruikersbeheer.....	28
Lidmaatschappen en Betalingen.....	28
Meertalige Ondersteuning.....	28
2. Thema's/ templates.....	28
3. Uitbreidingen.....	29
Plugins.....	29
Hooks.....	29
Authenticatie Providers.....	29
4. Bouwen en gebruik maken van RESTful API's.....	30
Eenvoudige Beheerdersauthenticatie.....	30
Uitgebreide Documentatie.....	30
5. Meertalige ondersteuning.....	31
6. Versiebeheer met GIT.....	32
7. Implementatie en onderhoud.....	33
8. Beveiliging.....	34
Conclusie:.....	35
Ghost CMS:.....	35
Sanity:.....	35
Strapi:.....	35

Introductie

In dit onderzoek paper gaan we dieper in op twee headless Content Management Systemen en onderzoeken we hun capaciteit op verschillende belangrijke gebieden. We zullen kijken naar installatiemethoden, configuratie opties, ondersteuning voor thema's en templates, mogelijkheden om uit te breiden via externe API's, het bouwen en gebruiken van RESTful API's, meertalige ondersteuning, versiebeheer met GIT en de implementatie van CI/CD-Pipelines, implementatie en onderhoud en mogelijke kwetsbaarheden van de CMS.

Dit onderzoek heeft als doel het identificeren van het ideale CMS voor ons project genaamd "VestIZ", waarbij we streven naar het maken van een slimmere en veiligere vestiaire. Het identificeren van een ideale CMS is essentieel, zodat we vanuit één centrale omgeving onze content kunnen beheren en die content kunnen publiceren op verschillende kanalen.

Aan het einde van dit onderzoek paper zullen we een weloverwogen keuze maken voor het headless CMS-systeem dat het beste aansluit bij de behoeften van ons VestIZ-project. Het doel van het CMS-systeem in ons project is om aankondigingen te beheren en bij te houden die door de evenement beheerders worden verzonden. Deze aankondigingen zullen zichtbaar zijn op de mobiele app van de gasten en op de televisies die in de zaal zijn geplaatst.

Voor ons is het vooral belangrijk dat de service gratis kan gebruikt worden en dat het admin paneel eenvoudig is, aangezien onze use-case vrij simpel is. Het is ook belangrijk dat we onze eigen data types gemakkelijk kunnen definiëren.

SANITY

Onderzoek

Sanity

Installatie

Stap 1: Node.js en npm installatie

Zorg ervoor dat node.js en npm geïnstalleerd zijn.

Stap 2: Sanity CLI installatie

Open CMD en voer het volgende commando uit om de Sanity te installeren

```
npm install -g @sanity/cli
```

Stap 3: Initialiseer een nieuw project

Ga naar de directory waar u uw Sanity project wilt aanmaken met behulp van CMD. Voer het volgende commando uit.

```
Sanity init
```

Stap 4: Aanmelden met Sanity.io & Configuratie

Er wordt gevraagd om u aan te melden met uw Sanity account. Volg hierna de instructies om uw project te configureren. U kunt kiezen voor een leeg template of een “starter” template.

Stap 5: Start de lokale ontwikkelings server

Voer het volgende commando uit om de lokale ontwikkelings server te starten.

```
Sanity start
```

Stap 6: Start Sanity studio

Navigeer naar de aangemaakte ‘studio’-directory en voer het volgende commando uit.

```
Sanity start
```

U kunt nu Sanity Studio openen in uw webbrowser. Standaard wordt dit gedaan op: <http://localhost:3333>. Log in met uw Sanity account.

Configuratie

Een van de kernsterkte van Sanity is de flexibele configuratie mogelijkheid voor inhoudstypen en aangepaste velden. Deze configuratie maakt het mogelijk om de structuur en inhoud van uw CMS volledig aan te passen aan de specifieke behoeften van uw project.

Binnen Sanity kunt u verschillende inhoudstypen definiëren, zoals "Blogberichten", "Producten", "Auteurs", of elk ander type content dat relevant is voor uw project. Deze inhoudstypen fungeren als de sjablonen voor uw content items. U kunt inhoudstypen maken en aanpassen via de configuratiebestanden in uw Sanity-project.

Voor elk inhoudstype kunt u aangepaste velden definiëren. Deze velden kunnen variëren van eenvoudige tekstvelden tot complexe relaties en afbeeldingenvelden. U kunt zelfs aangepaste regels instellen om ervoor te zorgen dat de ingevoerde gegevens voldoen aan uw vereisten.

Een andere handige functie van Sanity is de mogelijkheid om herbruikbare componenten te definiëren. Dit betekent dat u complexe inhoudselementen kunt maken en deze opnieuw kunt gebruiken in verschillende delen van uw project.

U kunt ook toegangscontrole instellen om te bepalen wie toegang heeft tot welke delen van uw CMS en welke bewerkingen ze kunnen uitvoeren. Dit is cruciaal voor het waarborgen van de veiligheid en integriteit van uw content.

Thema's/ Templates

Om thema's en sjablonen in Sanity Studio te gebruiken, begint u met het installeren van een thema, bijvoorbeeld het standaardbasis-thema, met het commando.

```
sanity install @sanity/base-theme
```

Daarna maakt u aangepaste sjablonen met 'Sanity init template mijn-sjabloon'. U configureert deze thema's en sjablonen vervolgens in het 'sanity.json'-bestand in de 'studio'-directory. Eventuele aanpassingen aan de thema's en sjablonen kunnen worden gemaakt door de relevante bestanden te bewerken. Vervolgens kunt u Sanity Studio starten met 'Sanity start' en toegang krijgen tot de aangepaste thema's en sjablonen in uw beheerinterface. Dit stelt u in staat om de styling en lay-out van uw Sanity Studio aan te passen aan uw specifieke projectvereisten.

Uitbreidingen

In Sanity zijn er verschillende manieren om de functionaliteit van je project op maat te maken. Dit omvat het integreren van externe plugins en integraties om extra functies zoals sociale media koppelingen, betalingsgateways en zoekmogelijkheden toe te voegen. Je hebt ook de vrijheid om aangepaste code toe te voegen als je technische vaardigheden hebt. Dit stelt je in staat om HTML, CSS en JS aan te passen en op maat gemaakte widgets, functies of lay-outwijzigingen te creëren om je website of applicatie te personaliseren.

Daarnaast kun je binnen Sanity de structuur van je content aanpassen aan de specifieke behoeften van je project. Dit omvat het configureren van verschillende contenttypen en aangepaste velden om de gegevens te structureren zoals jij dat wilt.

Bovendien biedt Sanity meertalige ondersteuning, waardoor je content kunt vertalen en lokaliseren om je publiek wereldwijd te bereiken.

Ten slotte kun je beveiligingsmaatregelen implementeren om de toegang tot je project te beheren en de veiligheid en integriteit van je content en gebruikersgegevens te waarborgen. Met deze mogelijkheden kun je Sanity aanpassen en uitbreiden om te voldoen aan de unieke vereisten van je project.

Bouwen en gebruik maken van RESTful API's

Door RESTful API's in Sanity te gebruiken, kunnen we speciale toegangspunten creëren om gegevens op te halen en te delen met andere software. Dit maakt het gemakkelijker om inhoud van Sanity te gebruiken in bijvoorbeeld mobiele apps of externe websites. Het gebruik van RESTful API's in Sanity vergroot de flexibiliteit van ons CMS en stelt ons in staat om content te integreren met andere tools en services. Hierdoor kunnen we contentbeheer aanpassen aan de behoeften van elk project en onze CMS-functionaliteit verbeteren. Dit is een waardevolle ontdekking voor het optimaliseren van Sanity in moderne digitale toepassingen.

Meertalige ondersteuning

Een van de krachtige kenmerken van het Sanity contentmanagementsysteem (CMS) is de ingebouwde ondersteuning voor meertaligheid. Dit stelt gebruikers in staat om content te maken en te beheren in meerdere talen binnen dezelfde CMS-omgeving. Deze functie is bijzonder waardevol voor projecten met een internationaal publiek of voor bedrijven die zich richten op verschillende markten over de hele wereld.

In Sanity kunnen ontwikkelaars en contentbeheerders eenvoudig meerdere talen instellen en beheren voor hun inhoud. Dit wordt gedaan door de creatie van specifieke taalversies van elk stuk content. Elk stuk content, zoals een artikel of een pagina, kan meerdere taalversies hebben die parallel worden beheerd. Hierdoor kunnen gebruikers de inhoud vertalen en lokaliseren voor verschillende doelgroepen.

Versiebeheer met GIT en de implementatie van CI/CD-Pipelines

U kunt Sanity automatisch hosten met CI. Dit is handig voor het automatisch bijwerken van de gehoste Studio wanneer u uw lokale wijzigingen naar bronrepositories pusht of handmatige releases uitvoert. Voeg @sanity/cli toe als een ontwikkelingsafhankelijkheid en configureer je CI/CD-workflow om het commando Sanity deploy uit te voeren. Vergeet niet om het bestand sanity.cli.js config in je studiomap te hebben. Je moet ook een autorisatietoken opgeven met de SANITY_AUTH_TOKEN omgevingsvariabele. Dit komt doordat Sanity de lokale gebruikerssessie gebruikt voor authenticatie met onze hosting service, die niet noodzakelijk beschikbaar is in je CI/CD workflow. Je kunt een deploy token aanmaken in het projectmanagement dashboard.

(bron: <https://www.sanity.io/docs/deployment>)

Implementatie en onderhoud

Het implementeren van Sanity omvat verschillende essentiële stappen, te beginnen met het creëren van een nieuw Sanity-project. Dit kan worden gedaan via de Sanity CLI of de Sanity Studio, die een aanpasbare beheerinterface biedt. Hier definieert u het schema van uw project, contenttypen en aangepaste velden om uw gegevens te structureren. Vul uw project met content, ofwel via de Sanity Studio of programmatisch via API's, waarbij u tekst, afbeeldingen en gestructureerde gegevens organiseert zoals nodig. Pas de Sanity Studio aan om bij uw projectbranding en UI-voorkeuren te passen, en maak aangepaste sjablonen om content op het front-end te presenteren. Eenmaal geïmplementeerd, is effectief onderhoud essentieel.

Werk regelmatig content bij en beheer deze binnen de Sanity Studio, houd deze nauwkeurig en actueel. Handhaaf beveiliging door potentiële kwetsbaarheden aan te pakken, ongeautoriseerde toegang te beperken en gegevensbescherming te waarborgen.

Monitor en optimaliseer voortdurend de prestaties, waarbij u zich richt op query's, afbeeldingen en contentlevering. Implementeer een robuuste back-upstrategie, plan voor gegevensherstel en schaal de infrastructuur op naarmate uw project groeit.

Blijf Git gebruiken voor versiebeheer om codeaanpassingen bij te houden, documenteren en integreren. Deze stappen zorgen voor een succesvolle implementatie en lopend onderhoud van uw Sanity-project, waardoor het betrouwbaar en aanpasbaar blijft aan uw behoeften.

Beveiliging

Sanity biedt organisatie- en projectbeheerders de mogelijkheid om granulaire toegangscontroles tot content op te geven. Gebruikers kunnen een rol toegewezen krijgen, die een reeks rechten definieert voor content die in de Sanity studio wordt bewerkt. Afhankelijk van uw plan kunt u ook gebruikers aanmelden via ingebouwde SSO-mogelijkheden. Hieronder vindt u handleidingen voor beide toegangsfuncties, samen met een volledige API-referentie voor het specificeren van rechten voor inhoud op basis van onze GROQ-querytaal.

CORS (Cross-Origin Resource Sharing) is een beveiligingstechniek die wordt gebruikt om het strikte webbrowserbeleid voor vertrouwde domeinen te versoepelen. Hierdoor kunnen bepaalde webpagina's verzoeken doen aan en toegang krijgen tot gegevens van een API op een ander domein.

Het Same-origin beleid van de webbrowser voorkomt normaal gesproken dat webpagina's verzoeken doen aan domeinen die niet dezelfde oorsprong hebben (URI-schema, hostnaam en poort). Dit beleid is cruciaal voor de beveiliging, omdat het voorkomt dat kwaadwillende websites toegang krijgen tot gevoelige gegevens op andere websites waarop je bent ingelogd, zoals je bankrekening.

Dit beleid kan echter een uitdaging vormen wanneer je services zoals Sanity gebruikt, waarbij je Content Studio of app voor één pagina op een andere oorsprong draait dan de API. CORS biedt redding door vertrouwde pagina's in staat te stellen dit beleid te omzeilen.

Het werkt als volgt: Wanneer een script op een webpagina gegevens van een ander domein probeert op te vragen, stuurt de browser een preflightverzoek naar de server, inclusief informatie over de oorsprong van de aanvragende pagina. Als de oorsprong vertrouwd is voor dat project, staat de server het verzoek toe. Anders wordt het geweigerd.

Daarnaast kunnen CORS-instellingen aangeven of er credentials (zoals sessiecookies of autorisatietokens) mogen worden meegestuurd met het verzoek. Één het bouwen van een app met één pagina die direct is verbonden met de API, zou je het verzenden van referenties voor de veiligheid meestal weigeren.

Houd in gedachten dat CORS alleen van toepassing is op browserbeveiliging en geen invloed heeft op aanvragen die buiten de browser worden gedaan, zoals in Node.js of met tools als curl.

Om data te wijzigen via de API, heb je een toegangstoken nodig met schrijfrechten, die afkomstig kunnen zijn van de browsersessie van de gebruiker of toegevoegd kunnen worden in de instellingen van het project. Openbare datasets zijn standaard alleen-lezen en vereisen schrijftoegang van uitgenodigde gebruikers of tokens.

(bron: <https://www.sanity.io/docs/access-control>,
<https://www.sanity.io/docs/browser-security-and-cors>)

Conclusie



1. Installatie

Het installeren van Ghost CMS lokaal is een eenvoudig proces. In dit gedeelte van de paper zal ik de stappen uiteenzetten om Ghost CMS lokaal te installeren en hoe deze taak relatief moeiteloos kan worden volbracht. We kiezen voor een lokale installatie omdat deze volledig kosteloos is. Een cloud optie is ook mogelijk, maar deze is helaas niet gratis te gebruiken.

Stappen voor de Lokale Installatie van Ghost CMS

Stap 1: Voorbereiding van de Omgeving

Voordat u Ghost CMS lokaal kunt installeren, moet u ervoor zorgen dat uw systeem aan bepaalde vereisten voldoet. De belangrijkste vereisten zijn Node.js en npm (Node Package Manager). U kunt deze tools gratis downloaden en installeren vanaf de officiële websites van Node.js.

Stap 2: Download Ghost CLI

Ghost biedt een handige Command Line Interface (CLI) waarmee u eenvoudig Ghost CMS kunt installeren en beheren. Om de Ghost CLI te installeren, opent u een terminal of opdrachtprompt en voert u het volgende commando in: `npm install -g ghost-cli`

Stap 3: Maak een Nieuwe Directory

Navigeer naar de directory waar u uw lokale Ghost CMS-installatie wilt opslaan. Maak een nieuwe directory met de naam van uw keuze en navigeer naar deze directory in de terminal.

Stap 4: Installeer Ghost

Voer het volgende commando uit om Ghost CMS lokaal te installeren: `ghost install local`

Dit commando zal de nieuwste versie van Ghost downloaden, de benodigde afhankelijkheden installeren en een SQLite-database instellen voor lokale ontwikkeling.

Stap 5: Configureer Ghost

Tijdens het installatieproces wordt u gevraagd om verschillende configuratie-opties in te voeren, zoals de URL van uw site en uw database-instellingen. Volg de instructies op het scherm om deze configuratie af te ronden.

Stap 6: Start Ghost

Zodra de installatie is voltooid en de configuratie correct is ingesteld, kunt u Ghost CMS lokaal starten met het volgende commando: `ghost start`

Dit zal de lokale ontwikkelingsserver van Ghost activeren, en u kunt uw site bekijken in uw webbrowser op de opgegeven URL, meestal <http://localhost:2368>.

2. Configuratiemogelijkheden

Ghost CMS biedt een uitgebreide reeks configuratiemogelijkheden waarmee gebruikers de functionaliteit en uitstraling van hun website volledig kunnen aanpassen aan hun behoeften en voorkeuren. Hieronder volgt een beknopt overzicht van enkele van de belangrijkste configuratiemogelijkheden binnen Ghost CMS:

Thema's en Ontwerpaanpassingen

Ghost ondersteunt thema's (themes) waarmee gebruikers het uiterlijk van hun website kunnen personaliseren. Er zijn zowel gratis als premium thema's beschikbaar. Gebruikers kunnen ook CSS aanpassen om hun website verder te stylen.

Mail

Je kunt onder andere de SMTP-serverinstellingen aanpassen voor e-mailverzending, het afzenderadres definiëren en zelfs aangepaste e-mailtemplates maken en bewerken voor verschillende gebeurtenissen en notificaties binnen het systeem. Daarnaast is het mogelijk om de frequentie en inhoud van automatische nieuwsbrieven in te stellen en te personaliseren voor je publiek.

Publicatie-instellingen

Gebruikers kunnen de publicatie-instellingen van hun berichten beheren, zoals de datum en tijd van publicatie, de URL-structuur, de weergave van auteurs en tags, en de mogelijkheid om berichten als concepten op te slaan.

Gebruikersbeheer

Ghost biedt verschillende niveaus van gebruikersrollen, waaronder beheerders, redacteurs, auteurs en leden. Dit stelt website-eigenaren in staat om meerdere medewerkers toegang te geven en hun rechten te beheren.

Integraties

Ghost ondersteunt integraties met externe diensten en platforms, zoals nieuwsbriefproviders, analytics-tools en sociale mediakanalen. Dit maakt het mogelijk om content te delen en te distribueren via verschillende kanalen.

Lidmaatschappen en Betalingen

Ghost biedt de mogelijkheid om lidmaatschapsprogramma's op te zetten en betalingsintegraties te configureren, waardoor website-eigenaren inkomsten kunnen genereren door exclusieve inhoud aan te bieden.

Meertalige Ondersteuning

Ghost ondersteunt meerdere talen en biedt de mogelijkheid om meertalige websites te maken met verschillende URL-structuren.

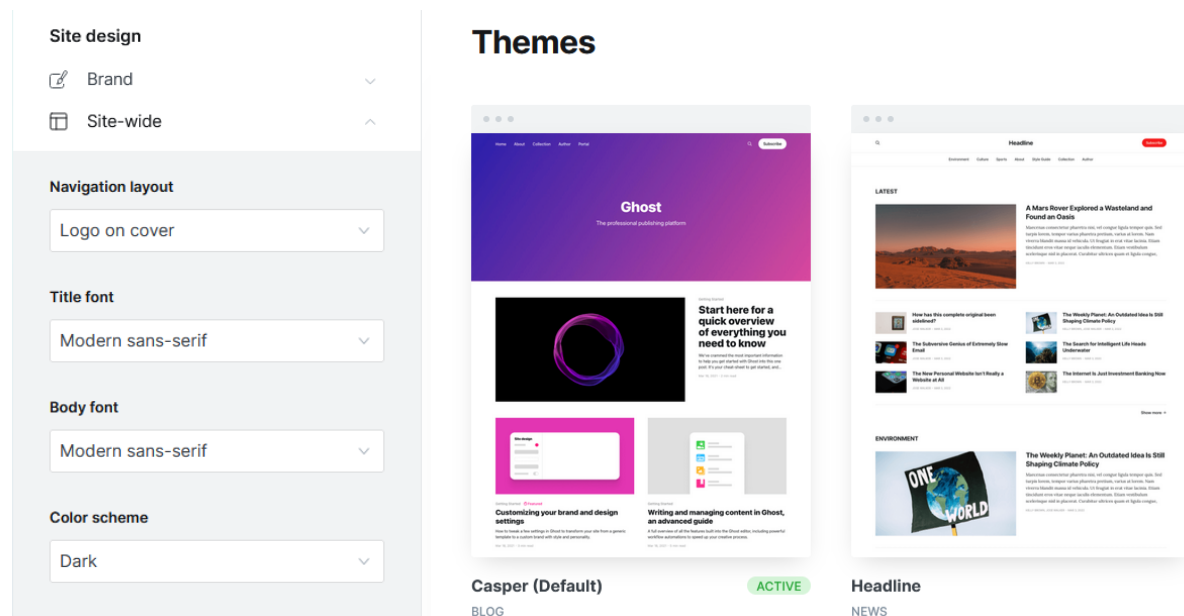
2. Thema's/ templates

Thema's kiezen en aanpassen

Je kunt kiezen uit verschillende thema's voor je Ghost-webpage en deze aanpassen aan je eigen smaak. Dit omvat het wijzigen van kleurenschema's, lettertypen, achtergronden en lay-outinstellingen. Sommige thema's bieden ook aanpassingsmogelijkheden voor de startpagina en individuele pagina's.

Templates bewerken

Ghost CMS maakt gebruik van Handlebars, een templatingtaal, om de structuur van je websitepagina's te beheren. Je kunt de templates bewerken om de weergave van specifieke pagina's aan te passen. Hiermee kun je de inhoudsindeling wijzigen, aangepaste widgets toevoegen en de HTML en CSS aanpassen.



3. Uitbreidingen

In Ghost CMS kun je uitbreidingen, zoals apps en integraties, aanpassen en configureren.

Apps en integraties installeren

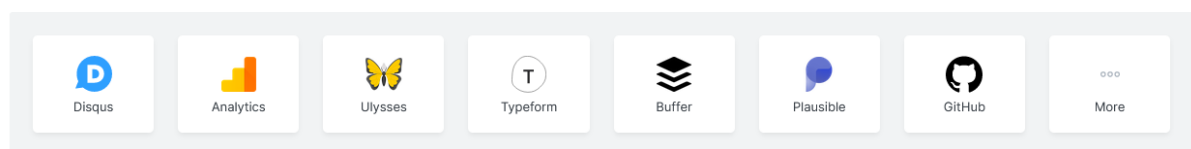
Je kunt verschillende apps en integraties van derden installeren om de functionaliteit van je Ghost-website uit te breiden. Dit omvat zaken als nieuwsbriefintegraties, socialemediakoppelingen, analysehulpmiddelen en meer.

Aangepaste code toevoegen

Als je technische kennis hebt, kun je aangepaste code toevoegen aan je Ghost-website om specifieke functies te implementeren. Dit omvat het bewerken van de HTML-, CSS- en JavaScript-code om aangepaste widgets, functies of lay-outaanpassingen te maken.


Settings > Integrations

Integrations



Settings > Code injection

Code injection

 Ghost allows you to inject code into the top and bottom of your theme files w for quick modifications to insert useful things like tracking codes and meta ta

Site Header

Code here will be injected into the `{{ghost_head}}` tag on every page of the site

1	
---	--

4. Bouwen en gebruik maken van RESTful API's

Content en Admin API's

Een opvallend aspect van Ghost CMS is de Content en Admin API. Deze bieden gebruikers de mogelijkheid om content te beheren vanuit vrijwel elke applicatie die HTTP-verzoeken kan verzenden. Dit opent de deur naar aangepaste workflows en naadloze integraties met andere tools en platforms, waardoor contentbeheerders hun creativiteit de vrije loop kunnen laten.

Gebruiksvriendelijke Libraries

Ghost CMS biedt gebruiksvriendelijke libraries en SDK's voor verschillende programmeertalen, waaronder JavaScript, Python en Ruby. Deze libraries zijn ontworpen met de nadruk op eenvoud en consistentie, waardoor ontwikkelaars snel aan de slag kunnen zonder tijdrovende documentatie te doorgronden.

Eenvoudige Beheerdersauthenticatie

Ghost CMS maakt authenticatie voor beheerders eenvoudig. Kopieer gewoon je 'key' vanuit de 'Integrations pagina' en stuur deze mee met je request. Dit vereenvoudigt het inlogproces en biedt een veilige manier voor beheerders om posts toe te voegen, aan te passen, verwijderen en zoveel meer.

Uitgebreide Documentatie

Ghost CMS onderscheidt zich ook door zijn uitgebreide en begrijpelijke documentatie. Deze bron biedt duidelijke uitleg, gedetailleerde voorbeelden en gebruiksscenario's, waardoor gebruikers snel antwoorden kunnen vinden en problemen kunnen oplossen.

Praktische voorbeelden

Posts opvragen

```
const api = new GhostContentAPI({
  url: BLOG_URL,
  key: CONTENT_API_KEY,
  version: "v5.0",
});

api.posts
  .browse({
    limit: "all",
    include: "tags,authors,html",
  })
  .then((posts) => {
    posts.forEach((post) => {
      console.log(post);
    });
  })
  .catch((err) => {
    console.error(err);
  });
```

Posts aanmaken

```
const api = new GhostAdminAPI({
  url: BLOG_URL,
  key: CONTENT_API_KEY,
  version: "v5.0",
});

api.posts.add({
  title: 'My first draft API post',
  html: "<p>My post content. Work in progress...</p>",
});
```

5. Meertalige ondersteuning

Ghost CMS biedt meertalige ondersteuning voor websites, maar deze functionaliteit kent enkele beperkingen. Het vertalen van je website is mogelijk, maar helaas niet in real-time.

Je moet van tevoren aangeven in welke talen je website beschikbaar moet zijn, wat flexibiliteit beperkt.

Bovendien vereist het handmatig aanmaken van 'locale' JSON-bestanden per taal aanzienlijke inspanning en is het potentieel foutgevoelig. Dit kan frustrerend zijn voor gebruikers die snel en gemakkelijk meertalige content willen beheren. Het is belangrijk om deze beperkingen te overwegen bij het kiezen van Ghost CMS voor meertalige websites.

Als je van plan bent om verschillende inhoud in elke taal te publiceren, raad Ghost we aan om één installatie per taal te gebruiken. Volgens hen is dit de manier die het gemakkelijkst op te zetten is, maar ook op de lange termijn het meest duurzaam is. Persoonlijk vind ik dit een omslachtige manier om meertalige ondersteuning te bereiken.

Dit is teleurstellend, gezien mijn overwegend positieve en moeiteloze ervaringen met het platform. Het voelt als een onnodige hindernis in een overigens soepel proces.

Voorbeeld Spaans locale json bestand:

```
{
  "Back": "Volver",
  "Newer Posts": "Artículos Siguientes",
  "Older Posts": "Artículos Anteriores",
  "Page {page} of {pages}": "Página {page} de {pages}",
  "Subscribe": "Suscríbete",
  "Subscribe to {blogtitle}": "Suscríbete a {blogtitle}",
  "Subscribed!": "¡Suscrito!",
}
```

6. Versiebeheer met GIT

De integratie met GitHub Actions biedt een krachtige oplossing voor het automatiseren van themadeploymentprocessen. Met GitHub Actions kunt u eenvoudig uw thema uploaden naar de Ghost Admin API. Hier is een beknopte handleiding om aan de slag te gaan:

- Maak een aangepaste integratie in Ghost Admin genaamd 'GitHub Actions'.
- Kopieer en plak je integratie details in je Github repository 'secrets'
Settings → Secrets and variables → Actions.
- Installeer de 'Ghost Theme Deploy Action' door onderstaande code in een nieuw bestand met de naam '.github/workflows/main.yml' te plaatsen.

```
name: Deploy Theme

on:
  push:
    branches:
      - master
      - main

jobs:
  deploy:
    runs-on: ubuntu-22.04

    steps:
      - uses: actions/checkout@v3
      - name: Deploy Ghost Theme
        uses: TryGhost/action-deploy-theme@v1
        with:
          api-url: ${{ secrets.GHOST_ADMIN_API_URL }}
          api-key: ${{ secrets.GHOST_ADMIN_API_KEY }}
```

- Elke keer dat u wijzigingen naar uw repository pusht, zal uw thema automatisch worden gedeployed naar Ghost Admin.

Zorg ervoor dat u in Ghost Admin naar 'Instellingen' → 'Thema' gaat om te controleren of het thema dat u vanuit GitHub uploadt, het actieve thema is. Met deze configuratie kunt u uw Ghost-thema eenvoudig beheren met GitHub Actions.

Dit is dus wel beperkt tot enkel het thema van uw CMS.

7. Security

Automatische SSL

De CLI-tool van Ghost probeert standaard automatisch SSL-certificaten te configureren voor alle nieuwe Ghost-installaties met Let's Encrypt

Standaardtoestemmingen

Ghost-CLI draait niet als root en configureert automatisch alle machtigingen van de servermap correct volgens de OWASP-standaarden.

Brute force protection

Inlogpogingen van gebruikers en aanvragen voor het resetten van wachtwoorden zijn allemaal beperkt tot 5 per uur per IP.

Data validatie and serialisatie

Ghost voert sterke serialisatie en validatie uit op alle gegevens die in de database worden ingevoerd, evenals geautomatiseerde symlink-bescherming op alle geüploade bestanden.

Encoded tokens

Alle gebruikersuitnodigingen en wachtwoord reset tokens zijn base64 gecodeerd met serverside geheim. Alle tokens zijn altijd voor eenmalig gebruik en verlopen altijd.

Wachtwoord hashing

Ghost volgt de verificatiestandaarden van OWASP en alle wachtwoorden worden gehasht en 'salted' met bcrypt om de integriteit van wachtwoorden te waarborgen.

SQLi preventie

Ghost gebruikt Bookshelf ORM + Knex query builder en genereert geen eigen ruwe SQL-query's. Ghost heeft geen interpolatie van variabelen direct naar SQL-strings.

XSS preventie

Ghost gebruikt overal 'escaped strings' om XSS te voorkomen



1. Installatie

Het installeren van Strapi CMS lokaal is heel Eenvoudig en gebruiksvriendelijk voor iedereen. In dit gedeelte zal ik uitleggen hoe Strapi lokaal geïnstalleerd moet worden.

Stappen voor het maken van een project met Strapi

Stap 1: Voorbereiding van de Omgeving

Vooraleer je Strapi kan installeren zijn er bepaalde voorwaarden. Node.js en npm (of yarn) zijn een vereiste om kunnen werken met Strapi. Als je voor de optie kiest om SQLite te gebruiken zal je hiervoor ook Python nodig hebben.

Stap 2: Download Strapi/ maak een nieuw project

Strapi is eenvoudig te installeren doormiddel van de Command Line Interface(CLI). U doet dit door een terminal te openen of opdrachtprompt en het volgende commando uit te voeren: `npx create-strapi-app@latest my-project --quickstart`

“my-project” vervang je door de naam van je project.

Stap 3: registreer de eerste admin gebruiker

Na dat de installatie voltooid is zal je browser een nieuwe pagina openen waar je een admin account zal aanmaken en toegang zult krijgen tot de admin panel.

De server zal voor de eerste keer zich zelf opstarten en zal beschikbaar zijn op <http://localhost:1337>

Het admin panel zal beschikbaar zijn op <http://localhost:1337/admin>

2. Configuratiemogelijkheden

Strapi biedt veelzijdige configuratiemogelijkheden aan. Het succes van een content management systeem hangt sterk af van hoe goed het kan worden aangepast aan de specifieke behoeften van een project. Strapi heeft zich bewezen als een krachtig CMS-platform met uitgebreide configuratiemogelijkheden, waardoor gebruikers in staat zijn om een breed scala aan aanpassingen te maken om aan hun unieke eisen te voldoen. Hieronder vindt u een kort overzicht van enkele van de voornaamste configuratieopties beschikbaar in Strapi CMS:

Content Types

Een van de kernconfiguraties binnen Strapi is de mogelijkheid om aangepaste contenttypes te maken. Contenttypes definiëren de structuur van de gegevens die kunnen worden opgeslagen en beheerd in het CMS. Strapi biedt een intuïtieve interface om deze contenttypes te definiëren, waarbij gebruikers attributen zoals tekst, afbeeldingen, datumvelden en zelfs complexe relaties tussen contenttypes kunnen specificeren. Dit stelt gebruikers in staat om de gegevensmodellering aan te passen aan de specifieke behoeften van hun project.

Bovendien maakt Strapi gebruik van een flexibel Content-First benadering, waarbij contenttypes eerst worden gedefinieerd en vervolgens worden gebruikt om content in te voeren. Dit stelt nieuwe gebruikers in staat om eenvoudig nieuwe inhoud te maken zonder dat er diepgaande technische kennis vereist is.

API-Configuratie

Een van de opvallende kenmerken van Strapi is de mogelijkheid om RESTful en GraphQL API's automatisch te genereren op basis van de geconfigureerde contenttypes. Dit biedt ontwikkelaars de flexibiliteit om de API-configuratie aan te passen aan hun projectbehoeften.

- **Authenticatie en autorisatie:** Strapi biedt diverse opties voor authenticatie en autorisatie, waaronder JWT (JSON Web Tokens), role-based toegangscontrole en integratie met externe authenticatiesystemen zoals OAuth. Deze functies stellen beheerders in staat om de beveiliging van hun applicatie te configureren op een manier die voldoet aan hun vereisten.
- **Endpoint-configuratie:** Ontwikkelaars hebben de mogelijkheid om aangepaste eindpunten te creëren, queryparameters te definiëren en zelfs complexe zoekopdrachten uit te voeren op de API. Dit stelt hen in staat om de API te finetunen voor optimale prestaties en gebruiksvriendelijkheid.

Plugin-Extenties

Strapi biedt een uitgebreide set van plugins waarmee gebruikers extra functionaliteit aan hun CMS kunnen toevoegen. Deze plugins omvatten alles, van mediabeheer tot zoekfunctionaliteit en e-commerce-integraties. Wat deze plugins nog krachtiger maakt, is de mogelijkheid om ze te configureren en aan te passen aan de specifieke vereisten van het project.

- **Mediabeheer:** De mediabeheerplugin stelt gebruikers in staat om afbeeldingen en andere media eenvoudig te beheren. Beheerders kunnen de opslaglocatie, de manier waarop bestanden worden geüpload en zelfs de grootte van afbeeldingen configureren voor automatische optimalisatie.
- **Internationalisatie:** Voor projecten die meertalige inhoud vereisen, biedt Strapi een internationalisatieplugin waarmee content in meerdere talen kan worden beheerd. Gebruikers kunnen de beschikbare talen configureren en vertalingen toevoegen op een intuïtieve manier.

Gebruikersbeheer

Strapi stelt beheerders in staat om verschillende gebruikersrollen te definiëren, zoals beheerders, auteurs en lezers, met bijbehorende rechten. Dit geeft controle over wie welke acties kan uitvoeren binnen het CMS.

Lidmaatschappen en Betalingen

Strapi biedt meerdere pakketten aan, van gratis tot betalend. Dit biedt de mogelijkheid om het beste pakket te kiezen die voor u het beste past.

Meertalige Ondersteuning

Strapi biedt robuuste ondersteuning voor meertalige websites, inclusief functies voor vertaling en lokalisatie.

2. Thema's/ templates

Strapi is een headless content management systeem (CMS) dat ontworpen is om de backend-infrastructuur van websites en applicaties te beheren. Hoewel Strapi zelf geen thema's of templates biedt zoals traditionele CMS-systemen, biedt het wel een flexibel raamwerk waarmee ontwikkelaars aangepaste contentstructuren en API's kunnen maken.

3. Uitbreidingen

Strapi is een zeer flexibel en aanpasbaar headless content management systeem (CMS). Het biedt een scala aan uitbreidingsmogelijkheden om de functionaliteit ervan aan te passen en uit te breiden.

Plugins

Strapi is een zeer flexibel en aanpasbaar headless content management systeem (CMS). Het biedt een scala aan uitbreidingsmogelijkheden om de functionaliteit ervan aan te passen en uit te breiden.

Hooks

Hooks in Strapi stellen ontwikkelaars in staat om aangepaste logica en gedrag toe te voegen aan verschillende gebeurtenissen binnen het CMS, zoals het opslaan van een record of het bijwerken van gegevens. Hiermee kunnen ontwikkelaars complexe workflows en bedrijfslogica implementeren.

Authenticatie Providers

Strapi ondersteunt verschillende authenticatieproviders, waaronder OAuth, JWT, en lokale gebruikersregistratie. Ontwikkelaars kunnen aangepaste authenticatieproviders integreren om de aanmeldingservaring aan te passen en te beveiligen.

4. Bouwen en gebruik maken van RESTful API's

Strapi is een CMS die alleen maar gebruik maakt van API call om alles te doen.

Eenvoudige Beheerdersauthenticatie

Strapi CMS vereenvoudigt het authenticatieproces voor beheerders. Met behulp van API-tokens of andere methoden kunnen beheerders veilig inloggen en toegang krijgen tot het beheerdersdashboard. Dit vereenvoudigt het beheren van content, zoals het toevoegen, aanpassen en verwijderen van gegevens, en biedt beheerders tegelijkertijd de benodigde beveiliging.

Uitgebreide Documentatie

Een ander opvallend aspect van Strapi CMS is de uitgebreide en begrijpelijke documentatie. Deze bron biedt duidelijke uitleg, gedetailleerde voorbeelden en praktische gebruiksscenario's, waardoor gebruikers snel antwoorden kunnen vinden en eventuele problemen kunnen oplossen. Dit draagt bij aan een soepele en efficiënte ervaring bij het werken met Strapi.

5. Meertalige ondersteuning

Strapi biedt een krachtige meertalige ondersteuning waarmee gebruikers moeiteloos meertalige contenttypes kunnen creëren. Dit betekent dat het mogelijk is om verschillende versies van uw inhoud te genereren voor verschillende talen of regio's, en deze naadloos te beheren. In Strapi heeft u de mogelijkheid om taalvariabelen te definiëren die specifiek zijn voor elke ondersteunde taal, waardoor u moeiteloos kunt schakelen tussen verschillende taalversies van uw inhoud.

Bovendien biedt Strapi uitbreidingsmogelijkheden in de vorm van plug-ins, waarmee u geavanceerde meertalige functies kunt toevoegen aan uw CMS. Hierdoor kunt u bijvoorbeeld automatische vertalingen implementeren of gebruikers de vrijheid geven om hun voorkeurstaal te kiezen.

De meertalige ondersteuning van Strapi vereenvoudigt het beheer van meertalige inhoud en stelt contentmakers in staat om hun boodschap wereldwijd te verspreiden. Dit alles zonder dat ze zich zorgen hoeven te maken over de technische obstakels die vaak gepaard gaan met het beheren van meertalige websites. Het opent de deur naar een bredere en diversere doelgroep, waardoor uw content en boodschap een wereldwijd bereik krijgen.

6. Versiebeheer met GIT

Strapi ondersteunt integratie met GitLab en andere versiebeheersystemen, waardoor ontwikkelaars hun Strapi-projecten effectief kunnen beheren en samenwerken. Deze integratie maakt het mogelijk om projectwijzigingen te volgen, moeiteloos samen te werken met teamleden en een geautomatiseerde ontwikkelingsworkflow te handhaven. Het gebruik van GitLab als versiebeheertool biedt structuur en efficiëntie in de ontwikkelingscyclus van Strapi-projecten.

(<https://docs.strapi.io/dev-docs/deployment/digitalocean#install-and-configure-git-versioning-on-your-server>)

7. Implementatie en onderhoud

Bij de implementatie van Strapi begint het proces met de configuratie van de software volgens de specifieke behoeften van het project. Dit omvat het definiëren van contenttypes, velden en relaties om de structuur van de gegevens te bepalen. Strapi biedt de flexibiliteit om aangepaste contenttypes te maken en past zich aan diverse projecten aan, van eenvoudige blogs tot complexe e-commerceplatforms.

Het onderhoud van Strapi is van cruciaal belang om de prestaties, beveiliging en betrouwbaarheid van de CMS te waarborgen. Dit omvat regelmatige updates van de Strapi-software en de bijbehorende plugins om beveiligingsproblemen op te lossen en nieuwe functies toe te voegen. Het monitoren van de prestaties en het oplossen van eventuele problemen, zoals traagheid of fouten, is ook een integraal onderdeel van het onderhoudsproces.

8. Beveiliging

Toegangscontrole is een belangrijk element van Strapi's beveiliging. Het systeem maakt gebruik van rollen en rechten om te bepalen welke gebruikers toegang hebben tot specifieke functies en content. Dit helpt bij het beperken van toegang tot gevoelige informatie.

Strapi ondersteunt diverse authenticatiemethoden, waaronder lokale gebruikersregistratie en tokens zoals JWT en OAuth. Dit biedt flexibiliteit bij het beheren van gebruikersidentiteit en -toegang.

Regelmatige beveiligingsupdates en patches worden uitgebracht om bekende kwetsbaarheden aan te pakken, en het is van vitaal belang dat beheerders deze updates tijdig implementeren om de beveiliging te handhaven.

Inputvalidatie is een ander belangrijk onderdeel van de beveiliging, waarmee het systeem kan detecteren en beschermen tegen aanvallen zoals SQL-injecties en cross-site scripting (XSS). De API-beveiliging van Strapi kan worden geconfigureerd om nauwkeurig de toegangsrechten tot API-eindpunten te beheren, inclusief welke gebruikers of toepassingen toegang hebben tot specifieke functies.

Wachtwoorden worden veilig opgeslagen met behulp van cryptografische hash-functies om ervoor te zorgen dat deze niet eenvoudig kunnen worden ontcijferd, zelfs niet in geval van een datalek.

Auditlogs registreren alle gebruikersacties, waardoor beheerders verdachte activiteiten kunnen opsporen en onderzoeken. Beheerders moeten ook zorgen voor de beveiliging van de server- en hostingomgeving waarin Strapi wordt uitgevoerd, inclusief firewall-configuratie, regelmatige software-updates en bescherming tegen DDoS-aanvallen.

In combinatie bieden deze beveiligingsmaatregelen een goede basis om gevoelige gegevens en content veilig te beheren binnen Strapi CMS. Het is echter essentieel dat beheerders actief blijven betrokken bij het handhaven van deze beveiligingslagen om mogelijke bedreigingen effectief te mitigeren.

Conclusie:

Ghost CMS:

Ghost CMS is een opmerkelijk interessant contentmanagementsysteem dat uitblinkt in gebruiksgemak en vrijheid bij het configureren van thema's en ontwerpmogelijkheden. De mogelijkheid om moeiteloos samen te werken met verschillende third-party software voegt een extra laag van flexibiliteit toe.

Echter, het is belangrijk dat Ghost CMS primair is ontworpen voor blogdoeleinden en, helaas, niet voldoet aan de specifieke eisen van ons project. Onze behoeften zijn het verzenden van korte aankondigingen binnen een specifieke categorie. In Ghost is het al complexer om aangepaste datatypes te creëren, terwijl andere CMS-systemen dit al standaard ondersteunen.

Sanity:

Sanity is een zeer interessante CMS. Projecten zijn simpel op te zetten en er is uitgebreide documentatie ter beschikking. Met een free tier heb je voor de meeste kleine projecten meer dan voldoende resources tot je beschikking, wat een groot pluspunt is.

Enkele zaken die (voor ons project) nadelen zijn: De Sanity UI is heel basic, je moet er heel wat aan gaan tweak en aanpassen. Als een niet-developer velden wilt toevoegen binnen Sanity, moet dat door middel van code wat niet altijd evident is. Sanity heeft een heel complexe leercurve, zeker als het de eerste keer is dat je een (headless) CMS gebruikt.

Strapi:

Strapi CMS is een uiterst veelzijdig en flexibel content management systeem dat gebruikers in staat stelt om moeiteloos websites en applicaties te ontwikkelen en te onderhouden. Het biedt een intuïtief beheerderspaneel, waarmee nieuwe datatypes met gemak kunnen worden gecreëerd en dit alles kan zonder kosten gedeployed worden. Dankzij deze indrukwekkende functionaliteiten vormt Strapi de ideale keuze voor ons project.