

Project:Automatisation de la création de sites e-commerce basés sur l'architecture JamStack en utilisant Gatsby et le CMS WordPress

Description:

Ce projet vise à automatiser la création des sites e-commerce en utilisant l'architecture JamStack avec Gatsby et le CMS WordPress. Pour y parvenir, nous allons mettre en place un script Node.js pour automatiser l'installation conteneurisée de WordPress sous Bedrock. De plus, nous allons créer un package npm pour faciliter la création de projets Gatsby qui représentent un modèle modulaire de base contenant toutes les composantes nécessaires au développement. Le résultat final sera une solution permettant de créer des sites e-commerce de manière reproductible et modulaire.

Outils: Node.js,php,GatsbyJs,Wordpress,woocommerce, docker-compose,Bedrock,GraphQL

Tâches principales:

Côté Wordpress:

1. Création d'un script node js qui permet d'automatiser l'installation conteneurisée de WordPress sous Bedrock.
2. Creation d'un theme Wordpress minimal:

Le thème minimal pour WordPress destiné à être utilisé avec Gatsby est un thème léger et simplifié qui doit offrir une mise en page minimale et une feuille de style de base. Cependant, il doit être facilement personnalisé pour répondre aux besoins spécifiques du site Web. Pour offrir une grande flexibilité, il est recommandé de créer également un thème enfant, ce qui permet de préserver l'intégrité du thème parent et d'apporter des modifications sans affecter les fichiers de base.

A noter : On va utiliser le compositeur Bedrock pour installer WordPress car le compositeur permet :

- De gérer WP comme dépendance de projet, c'est-à-dire que pour l'installer, il suffit juste de taper une seule commande, ce qui facilite l'automatisation de ce processus.
- D'avoir la possibilité d'installer les plugins directement à partir de Composer.
- D'avoir un contrôle total sur la version installée de WP et de leurs plugins, et de contrôler les mises à jour et les données susceptibles d'être modifiées avec un plugin.
- De sécuriser l'accès en limitant les permissions.

Les opérations à effectuer dans c'est phase

- mise en place du docker-compose
- configuration de docker
- installation de bedrock
- configuration de bedrock
- chargement d'une base de données
- suppression des thèmes & plugins par défaut de Wordpress
- installation des plugins nécessaires :
 - les plugins peuvent être installés via Composer et [WP Packagist](#)
 - Les plugins suivants sont très importants pour fournir au développeur et à l'utilisateur final un Back office Wordpress fonctionnel:
 - GravityForms
 - Advanced Custom Fields
 - autodescription (The SEO Framework)
 - duplicate-wp-page-post
 - gutenber-plugin
 - Custom TinyMCE
 - Il faut faire une recherche pour savoir si vous aurez besoin d'installer d'autres plugins.
 - Les plugins suivants sont nécessaires pour la gestion de la communication entre Gatsby et Wordpress:
 - wp-gatsby
 - wp-graphq
 - disable-responsive-images-complete
 - clean-image-filenames
 - wp-graphql-widgets
 - wp-graphql-the-seo-framework
 - wp-graphql-acf
- Ajouter la thème destiné à être utilisé avec Gatsby

Votre travail consistera à automatiser toutes ces opérations en utilisant un seul script codejs reposant sur l'utilisation de fenêtres de dialogue prompts pour poser des questions à l'utilisateur. En fonction de ces réponses, l'installation sera préconfigurée.

Exemple d'exécution de script d'installation

Bienvenue dans le programme automatique d'installation de Bedrock

- Nom du projet?
 - test
- Project base URL? (test.docker.localhost)
 - test.docker.localhost #au le site va être accessible
- Database name? (test)
 - test
- Database user? (test)
 - user1
- Database password? (test)
 - testpw
- Database root password? (test)
 -
- Précharger la base de données avec des données fictives ? (y/N)
 - y
- Ajouter woocommerce dans le projet? (y/N)
 - y

Les étapes incluent la saisie d'informations de base telles que le nom du projet, l'URL de base du projet, le nom de la base de données, l'utilisateur de la base de données et le mot de passe. Il est également possible de précharger la base de données avec des données fictives et d'ajouter WooCommerce au projet. mais c'est juste des propositions vous pouvez ajouter d'autres options.

Côté Gatsby:

1. La création d'un package npm destiné à être utilisé pour initialiser un nouveau site GatsbyJs:

Ce package a pour but de faciliter la création d'un projet Gatsby en fournissant tous les modèles (Templates), les pages, blocks, les plugins de Gatsby et les composants React nécessaires pour reconstruire toutes les données (pages, articles, produits, les fonctionnalités de woocommerce, images, champs personnalisés, etc.) récupérées depuis le CMS WordPress. voici en general les modèles, les pages, les plugins et les blocks de Gatsby nécessaires:

Modèles : Il est recommandé de créer des modèles pour chaque type de contenu récupéré depuis WordPress, par exemple, un modèle pour les pages, un modèle pour les articles de blog, un modèle pour les produits, produit individuel, les catégories de produits, les pages d'accueil de la boutique, etc. Ces modèles permettent de structurer les données de manière cohérente et d'appliquer un design commun à chaque type de contenu.

Pages : vous devrez créer des pages pour les différentes parties d'un site e-commerce, comme la page d'accueil de la boutique, les pages de catégories de produits, les pages de produits individuels, les pages de panier, etc.

Plugins: Pour importer les données depuis WordPress, on peut utiliser le plugin Gatsby-source-wordpress. Il permet de récupérer les données de WordPress et de les transformer en format compatible avec Gatsby. D'autres plugins Gatsby peuvent également être utilisés pour améliorer les performances du site, ajouter des fonctionnalités, etc.

Blocks : Les blocks sont des éléments de contenu modulaires dans WordPress qui peuvent être réutilisés dans plusieurs pages. Le plugin "Advanced Custom Fields permet de créer des blocs personnalisés pour WordPress. Ces blocs peuvent être ensuite récupérés dans Gatsby en utilisant le plugin "gatsby-source-wordpress".

Ce package doit avoir un simple script Node JS qui repose sur l'utilisation de prompts afin de poser des questions à l'utilisateur ; En fonction de ces réponses, l'installation est préconfigurée. On détermine la liste des dépendances à installer selon les réponses fournies à l'installateur.

Exemple d'exécution de script d'installation

\$ npm init @packageName

INSTALLATION GATSBY

- ✓ Nom du site ... test
- ✓ URL vers le GraphQL de la source (WP) ... http://mano.docker.localhost/wp/graphql
- ✓ URL du site final ... http://mano.docker.localhost/
- ✓ site multilangues ... off / on
- ✓ site e-commerce ... off / on
- ✓ Le site propose une fonctionnalité de recherche ... off / on
- ✓ Le site propose une visionneuse de PDF ... off / on #La visionneuse de PDF est une fonctionnalité qui permet aux utilisateurs de visualiser les fichiers PDF directement sur votre site web, sans avoir besoin de les télécharger ou d'utiliser un lecteur de PDF externe.
- ✓ Le site propose une carte de géolocalisation ... off / on
- ✓ Le site génère des pages de catégories enrichies ... off / on
- ✓ Le fil d'Ariane est placé dans le corps de la page ... off / on
- ✓ La phrase d'accroche du site est reprise dans le header ... off / on
- ✓ La génération des fonts depuis Google Fonts est automatisée ... off / on
- ✓ La génération des icônes depuis ./static/assets est automatisée ... off / on
- ✓ Dans la grille de navigation, le titre des images apparaît comme un bouton ... off / on
- ✓ On commence ? ... off / on

c'est juste des propositions vous pouvez ajouter d'autres options.

Création d'un site démo: À la fin de la phase d'automatisation, on va travailler sur la création d'un petit site e-commerce en utilisant notre solution. Nous discuterons de cette phase à la fin de l'automatisation

Remarque : Les tâches principales peuvent être mises à jour en fonction de l'avancement du projet.