Laravel : Sessions et Référencement

Ce Travail Pratique viens anticiper sur des besoins de Knowide par rapport au référencement de son site et permettre une meilleure gestion de sessions d'utilisateurs ; ce TP vous aidera dans vos projets à venir.

Le TP se décline en deux parties :

PARTIE 1 : Enregistrer les sessions des utilisateurs dans la Base de données

Ce TP est un guide pour enregistrer les informations de session des utilisateurs dans la base de données au lieu des fichiers.

#Introduction

Le protocole HTTP étant sans état, les sessions offrent un moyen de stocker les informations de l'utilisateur à travers les requêtes HTTP.

Par défaut, Laravel enregistre les informations de session des utilisateurs dans des fichiers. Nous pouvons le voir à travers le fichier config/session.php où la configuration des sessions est définie :

```
...
'driver' => env('SESSION_DRIVER', 'file')
...
```

Cette ligne montre que le "driver" (pilote) pour la session est renseigné par la variable "SESSION_DRIVER" au fichier .env avec "file" comme valeur par défaut.

En cherchant cette variable au fichier .env, nous trouvons cette ligne :

```
...
SESSION_DRIVER=file
...
```

Les informations des sessions sont donc enregistrées dans des fichiers que nous pouvons retrouver au répertoire storage/framework/sessions.

Hormis utiliser les fichiers pour sauvegarder les informations de session, Laravel propose d'autres moyens tels que les cookies, memcached, redis, … Nous allons voir au point suivant comment enregistrer ces informations dans la base de données.

#Les informations de session dans la base de données

Pour enregistrer les informations des sessions des utilisateurs dans la base de données, nous avons besoin de :

1. Créer la table "sessions" pour stocker les informations
2. Changer le driver de session

1. Générer la table des sessions

Ouvrez la console (l'invite de commandes) à la racine de votre projet, exécutez la commande suivante pour créer le fichier de migration de la table "sessions" (database/migrations/..._create_sessions_table.php) :

```
php artisan session:table
```

Ensuite la commande suivante pour migrer la table "sessions" :

```
php artisan migrate
```

La table "sessions" nous donne accès aux informations suivantes de l'utilisateur :

- son identifiant s'il est authentifié (user_id)
- son adresse IP (ip_address)
- le navigateur qu'il utilise (user_agent)
- les données de sa session dans payload
- le timestamp de sa dernière activité (last_activity)

Notons qu'il est possible d'ajouter d'autres attributs si on le souhaite

2. Changer le driver des sessions

Editez le fichier .env en modifiant la valeur "file" en "database" pour la variable "SESSION_DRIVER " :

```
...
SESSION_DRIVER=database
...
```

Et lorsque maintenant un utilisateur viendra sur l'application, nous aurons ses informations dans la table "sessions"

Avec les informations collectées, nous pouvons compter le nombre d'utilisateurs authentifiés, connaître les navigateurs qu'ils utilisent le plus, faire des statistiques, ...

PARTIE 2 : SEO Laravel - Gérer les meta tags avec Cagilo

L'objectif de cette partie est de pouvoir générer dynamiquement les éléments <title> et <meta> pour le SEO en utilisant le package cagilo/cagilo dans un projet Laravel

#Définition du SEO

Le SEO (Search Engine Optimization) est l'acronyme qui signifie « Optimisation pour les moteurs de recherche » en français. C'est un ensemble de techniques permettant de positionner un site web dans les moteurs de recherche afin de le rendre visible auprès des internautes. Également appelé « référencement naturel », le SEO permet d'augmenter la qualité et la quantité de trafic sur un site web grâce aux résultats de recherches organiques non payants.

Contrairement au SEA (Search Engine Advertising), les résultats organiques sont eux obtenus grâce à un référencement sans publicités, dont l'objectif est de répondre à la requête des internautes au sein des pages de résultats de recherches (SERP). Le SEO s'applique donc aux moteurs de recherche mais se réfère également aux besoins des internautes. Pour cela, il est important de comprendre les attentes des utilisateurs lorsqu'ils tapent une requête sur Google afin de leur apporter la meilleure réponse possible. Google étant le moteur de recherche le plus utilisé avec 92% des parts de marché dans le monde, il est préférable de s'intéresser à ce dernier pour bâtir une stratégie SEO.

D'une manière globale, le SEO permet de renforcer et d'améliorer la visibilité d'un site pour gagner des positions sur des requêtes généralistes et plus spécifiques recherchées par les internautes. L'objectif ultime du SEO est d'amener son site dans les meilleures positions des pages de résultats de recherches (SERP) pour répondre à un besoin utilisateurs et ainsi atteindre vos objectifs de trafic et de conversions. Pilier de l'inbound marketing, le SEO a pour mission d'attirer les visiteurs vers votre site web. C'est le principal levier d'acquisition de trafic d'un site. Selon l'étude de Brightedge, 51% du trafic d'un site web provient en moyenne du canal organique, loin devant les réseaux sociaux qui ne représentent que 5%.

pour apprendre plus sur SEO, visiter :

https://developers.google.com/search/docs/beginner/seo-starter-guide?hl=fr

#Introduction au meta tag

Un meta tag (ou métaélément, ou élément meta ou encore balise meta) est une information qui porte sur la nature et le contenu d'une page web. On l'insère dans l'en-tête <header> d'une page à travers la balise <meta> :

```
<meta name="propriété" content="valeur" />
```

Les balises <meta> jouent un rôle essentiel sur le référencement organique (SEO) d'une page web sur les moteurs de recherche et sa présentation sur les médias sociaux (Facebook, Twitter, LinkedIn, ...). Ils permettent de renseigner les informations telles que le titre, la description, l'URL canonique, l'image, les métadonnées Open Graph, ... d'une page web.

Nous voulons voir comment insérer dynamiquement les balises SEO <title> et <meta> (description et Open Graph) dans une page web en utilisant le package Cagilo ou cagilo/cagilo dans un projet Laravel.

#Installer et configurer Cagilo

Le package cagilo/cagilo est un kit de composants Blade open-source pour le framework Laravel. Parmi ses composants, on retrouve « Meta » qui permet d'insérer les balises <title> et <meta>, y compris les métadonnées Open Graph pour les médias sociaux.

Pour installer cagilo/cagilo, on exécute la commande composer suivante :

```
composer require cagilo/cagilo
```

cagilo/cagilo nécessite php 8 et Laravel 8 ou plus comme on peut le voir au fichier composer.json :

```
"php": "^8.0",
"laravel/framework": "^8.59",
```

Une fois Cagilo installé, nous pouvons publier son fichier de configuration cagilo.php en exécutant la commande artisan suivante :

```
php artisan vendor:publish --tag=cagilo
```

Cette commande importe le fichier /config/cagilo.php où nous pouvons référencer les composants Cagilo à charger dans l'application et le chemin d'icônes à utiliser :

```
<?php

use Cagilo\UI\Components;

return [
    // Les composants à charger dans l'application
    'components' => [
        // 'alert'  => Components\Alert::class,
        // 'device' => Components\Device::class,
        // 'error'  => Components\Error::class,
        // 'icon'   => Components\Icon::class,
        // 'logout' => Components\Logout::class,
        'meta'   => Components\Meta::class,
    ],

    /* Le chemin vers le dossier d'icônes SVG.
    Example: [ 'fa' => storage_path('app/fontawesome') ]
    */
    'icons' => [

    ],
];
```

Nous commentons les composants alert, device, error, icon et logout de Cagilo pour de raisons de performance et laissons le composant meta que nous allons utiliser.

Balises SEO title et meta avec Cagilo

Pour insérer dynamiquement les balises SEO <title> et <meta> (description, Open Graph, …) dans une page web, Cagilo propose la balise <x-meta /> à laquelle nous pouvons transmettre les données via les attributs HTML suivants :

- « title » : Le titre
- « description » : La description
- « image » : L'image de présentation
- « url » : L'URL canonique
- « keywords » : Les mots clés
- « robots » : Règle d'indexation pour les moteurs de recherche
- « author » : L'auteur…

Considérons que nous avons un objet $post (une publication) représentée par le modèle app/Models/Post.php. Pour insérer les balises <title> et <meta> avec les informations du $post (titre, description, image, auteur et url) sur une vue (template Blade), nous devons compléter la balise <x-meta /> de la manière suivante :

```
<x-meta
        title="{{ $post->title }}"
        description="{{ $post->description }}"
        image="{{ asset('storage/'.$post->picture) }}"
        author="{{ $post->user->name }}"
        url="{{ route('posts.show', $post) }}"
/>
```

Ce qui va générer le code HTML suivant pour le $post :

```
<!-- Le titre -->
<title>Laravel</title>

<!-- Balises méta principales -->
<meta name="title" content="Laravel">
<meta name="description" content="The PHP Framework for Web Artisans">

<!-- L'auteur -->
<meta name="author" content="TOYI samrou">

<!-- Open Graph / Facebook / LinkedIn -->
<meta property="og:type" content="website">
<meta property="og:url" content="http://exemple.test/posts/2"/>
<meta property="og:locale" content="en"/>
<meta property="og:title" content="Laravel"/>
<meta property="og:description" content="The PHP Framework for Web Artisans">
<meta property="og:image" content="http://exemple.test/storage/posts/2.png">

<!-- Twitter Card  -->
<meta name="twitter:card" content="summary_large_image"/>
<meta name="twitter:url" content="http://exemple.test/posts/2">
<meta name="twitter:title" content="Laravel">
<meta name="twitter:description" content="The PHP Framework for Web Artisans">
<meta name="twitter:image" content="http://exemple.test/storage/posts/2.png">
```