

PHP – Laravel : TIMESTAMP & LA GESTION DE DATES

Ce TP vient appui au cours sur la localisation vue précédemment et vous serez très important pour les deux équipes dans vos projets.

Le TP se décline en deux parties :

1 - PHP : Convertir un timestamp en date, heures, minutes et secondes

Cette partie vous guidera pour formater un timestamp en jour, mois, année, heures, minutes et secondes avec les fonctions PHP `date()` et `gmdate()`.

Introduction au timestamp

Un timestamp est un entier (integer) qui représente le nombre de secondes écoulées depuis l'époque Unix (1er Janvier 1970, 0h00 00s GMT).

Nous pouvons obtenir le timestamp actuel ou d'une date antérieur en utilisant les fonctions PHP `time()`, `date($format)` ou `strtotime($datetime)` :

```
<?php

// Timestamp actuel
echo time(); // -> 1630116380

// ou
echo date("U"); // -> 1630116380

// ou encore
echo strtotime("now"); // -> 1630116380

// Timestamp du 01 Aout 2021
echo strtotime("2021/08/01"); // -> 1627776000
```

Nous voulons voir dans ce guide comment transformer un timestamp en une date (jour/mois/année), heure, minutes, secondes en utilisant les fonctions `date()` et `gmdate()` qui sont disponible depuis la version 4 de PHP.

Les fonctions `date()` et `gmdate()`

Les fonctions `date($format, $timestamp)` et `gmdate($format, $timestamp)` permettent de formater un timestamp `$timestamp` au format `$format` fourni.

Les paramètres

`$format` doit respecter les options de formatage pour la fonction `date()`.

Si `$timestamp` n'est pas renseigné ou est null, sa valeur par défaut est le timestamp actuel : valeur de retour de la fonction `time()`.

`gmdate()` est identique à la fonction `date()`, hormis le fait que le temps retourné est GMT (Temps solaire moyen au méridien de Greenwich).

Exemple : Timestamp en date, heures, minutes et secondes

Formations un timestamp pour obtenir une date complète avec l'heure au format « Mercredi 01 Aout 2021 à 12:30:49 » avec les fonctions `gmdate()` et `date()` :

```
<?php

// Timestamp
$timestamp = 1641040215;

// Le format
$format = "l d M Y à H:i:s";

// Formatage en jour, mois, année, heure, minutes et secondes
echo date($format, $timestamp); // -> Saturday 01 Jan 2022 à 12:30:15

// ou encore
echo gmdate($format, $timestamp); // -> Saturday 01 Jan 2022 à 12:30:15
```

2 - Laravel : Modifier la langue de la librairie Carbon

guide pour changer la langue de la librairie Carbon qui gère les dates dans un projet Laravel.

Introduction

Carbon est une API PHP de gestion des dates qui hérite de la classe `DateTime`. Le framework Laravel l'intègre dans un projet pour caster (transtyper, transformer) par exemple les dates `created_at` et `updated_at` d'un modèle en objet Carbon. Ainsi, nous pouvons appeler les méthodes Carbon sur ces dates.

Au script ci-dessous, j'affiche la date `created_at` d'un "App\Post" `$post` de façon facilement lisible pour un être humain sur un template blade :

```
@php
$post_date = $post->created_at;
@endphp

<p>
{{ $post_date->dayName }} - {{ $post_date->isoFormat("LL") }} - {{ $post->created_at->diffForHumans() }}
</p>
```

J'obtiens "Monday - March 2, 2020 - 22 hours ago", tout est en anglais. Nous allons voir au point suivant comment traduire cette date en "lundi - 2 mars 2020 - il y a 23 heures" ou une autre langue de votre choix : anglais (en), Espagnol (es), Russe (ru), ...

Je vous recommande de parcourir la documentation de Carbon (<https://carbon.nesbot.com/docs/>) pour découvrir ses nombreuses fonctions.

Modifier la langue

Nous pouvons modifier la langue de la librairie Carbon dans un projet Laravel soit en modifiant la langue globale de l'application dans la fonction boot() du fichier app\Providers\AppServiceProvider.php :

```
<?php

namespace App\Providers;
use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    // ...

    public function boot()
    {
        // On défini le "français" comme langue globale de l'application
        \App::setLocale('fr');
    }
}
```

Ou en modifiant spécifiquement la langue de Carbon :

```
<?php

namespace App\Providers;
use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    // ...

    public function boot()
    {
        // On défini le "français" comme langue globale de Carbon
        \Carbon\Carbon::setLocale('fr');
    }
}
```

Il ne reste plus maintenant qu'à actualiser la page dans le navigateur pour voir la traduction.

Vous venez également de voir une directive de blade @php @endphp

Cette directive permet de contourner un peu le pattern MVC pour permettre au développeur de quelque traitement de données sur la vue (mais il ne faut pas en abuser)