

# Laravel : Créer un helper

**Auteur**

[Wilo Ahadi](#)

**Domaine**

[Développement d'app. web](#)

**Technologies**

[Laravel](#), [PHP](#)

Nous vous présentons une technique pour créer vos propres helpers (fonctions PHP) puis les importer dans un projet Laravel.

**Sommaire**

- [Qu'est-ce qu'un helper dans Laravel ?](#)
- [1. Créer un fichier d'helpers](#)
- [2. Importer le fichier d'helpers](#)

## Qu'est-ce qu'un helper dans Laravel ?

**Un helper** est une fonction PHP globale qu'on peut appeler n'importe où dans une application Laravel pour effectuer certaines opérations tels que générer le chemin d'un répertoire (`app_path()`, `public_path()`, ...) ou une URL (`route()`, `asset()`, ...), récupérer une information de configuration (`config()`), ...

Laravel propose [des nombreux helpers](#) pour travailler avec les tableaux et les objets, les chaînes de caractères, les requêtes et les réponses HTTP, ... Mais si ces fonctions ne répondent pas exactement à vos attentes, voici la technique pour créer les vôtres :

# 1. Créer un fichier d'helpers

Pour bien organiser le projet, nous vous recommandons de créer un sous-dossier **helpers** dans le dossier **/ressources** où vous pouvez placer tous les fichiers qui contiennent vos propres fonctions, disons « fichiers d'helpers ».

Pour ma part, je crée un fichier **user\_helpers.php** où je définis la fonction `getUserByEmail()` qui me permet de récupérer un utilisateur `App\User` à partir de son adresse email.

Le code source du fichier **/ressources/helpers/user\_helpers.php** :

```
<?php

use App\User;

// On vérifie si la fonction n'existe pas déjà
if (!function_exists("getUserByEmail")) {

    // On crée l'helper de récupération d'un App\User par une adresse email
    function getUserByEmail ($email) {

        $user = User::where("email", $email)->first();

        if ($user) { return $user; } else { return false; }

    }

}

?>
```

## 2. Importer le fichier d'helpers

Pour rendre les helpers accessibles dans l'application, il faut charger les fichiers où ils sont définis. Pour ce faire, voici 2 astuces (prenez celle qui vous convient) :

### Astuce 1

Au fichier **composer.json** à la racine du projet, renseigner le chemin du fichier d'helpers au tableau « files » de la propriété « autoload » :

```
...
"autoload": {
    "psr-4": {
        "App\\": "app/"
    },
    "classmap": [
        "database/seeds",
        "database/factories"
    ],
}
```

```
        "files" : [ "resources/helpers/user_helpers.php" ]
    }
    ...
```

Puis exécuter la commande composer suivante dans la console pour charger automatiquement le fichier :

```
composer dump-autoload
```

## Astuce 2

Dans la méthode `boot()` du fichier **App\Providers\AppServiceProvider**, inclure le fichier d'helpers :

```
...
public function boot()
{
    // Importation du fichier resources/helpers/user_helpers.php
    require_once resource_path("helpers/user_helpers.php");
}
...
```

Maintenant que le fichier d'helpers est chargé, il ne reste plus qu'à appeler ma fonction.

### Exemple

```
Route::get('/', function () {

    // Récupération de l'utilisateur par son email
    $user = getUserByEmail("wilo.ahadi@gmail.com");

    if ($user) {

        return view('welcome', compact("user"));

    } else { abort(404); }

});
```