

LARAVEL

Un framework qui rend heureux

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RECENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

ACCUEIL ▾

LARAVEL 8

ANCIENNES VERSIONS ▾

FRONT-END 🔗 🔍

LES CMS ET OUTILS

LARAVEL 5

Cours Laravel 5.5 – les données – les ressources

1 octobre 2017

Avec Laravel on peut créer facilement un routage et un contrôleur REST/CRUD, autrement appelé contrôleur de ressource. Il est aussi aisé de bâtir des API et de travailler avec du JSON. Dans ce chapitre je vous propose d'explorer cet aspect de Laravel.

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

Préparation

Pour ce chapitre on va partir d'une installation vierge de Laravel :

```
composer create-project --prefer-dist laravel/laravel laravel5
```

Complétée par l'authentification :

```
php artisan make:auth
```

Et par une migration (avec la bonne configuration dans le fichier **.env**) :

```
php artisan migrate
```

Vous devez ainsi avoir un Laravel fonctionnel :

Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

Et les tables dans la base :

Table ▲
migrations
password_resets
users
3 tables

On va utiliser **tinker** pour créer quelques utilisateurs :

```
php artisan tinker
```

```
Psy Shell v0.8.11 (PHP 7.1.9 - cli) b
Justin Hileman
```

```
>>> factory(App\User::class, 10)-
>create()
```

```
=>
Illuminate\Database\Eloquent\Collection {#765
  all: [
    App\User {#761
      name: "Rigoberto Balistreri
Sr.",
      email: "xstroman@example.com",
      updated_at: "2017-09-30
22:03:29",
      created_at: "2017-09-30
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois ▼

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```
22:03:29",
    id: 1,
  },
  ...
}
```

J’ai déjà parlé des factories dans un précédent chapitre.

On utilise alors la méthode **create** d’Eloquent pour enregistrer les données dans la base :

id	name	email	password	remember_token	created_at	updated_at
1	Roberto Balistreri Sr	xstroman@example.com	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	60byU0a4P	2017-09-30 22:03:29	2017-09-30 22:03:29
2	Noble Fisher Sr	okay63@example.com	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	gP8E4k4GI	2017-09-30 22:03:29	2017-09-30 22:03:29
3	Brenna Mohr	kirlin.arthur@example.net	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	OoCHMEc69k	2017-09-30 22:03:29	2017-09-30 22:03:29
4	Heleen Graham	larkin.leslie@example.net	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	Hq65evVZD9j	2017-09-30 22:03:29	2017-09-30 22:03:29
5	Gardner Trantow	hchamplin@example.net	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	AyLkKoyK9I	2017-09-30 22:03:29	2017-09-30 22:03:29
6	Adrianna Kautzer	hane.normal@example.com	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	jY75o7s3Vl	2017-09-30 22:03:29	2017-09-30 22:03:29
7	Eldado Douglas IV	kutch.johnson@example.net	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	IN4vYAOV9r	2017-09-30 22:03:29	2017-09-30 22:03:29
8	Susie Koch	leannon.rubie@example.net	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	qJVVb1t6Ux	2017-09-30 22:03:29	2017-09-30 22:03:29
9	Mr. Ned Harvey V	kemmer.tony@example.com	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	UDSL5uJG8u	2017-09-30 22:03:29	2017-09-30 22:03:29
10	Kareem Hittch	hermosa74@example.org	\$2y\$10\$BjDdFfFa o12rttlaAFH1OpQy1d8t2LQyVleVRMzO...	BNW4UGL7N	2017-09-30 22:03:29	2017-09-30 22:03:29

Vous voyez ce qu’on peut faire avec juste une ligne dans **tinker** !

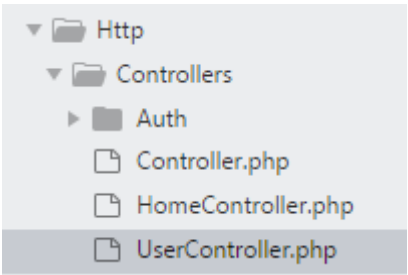
Les contrôleurs de ressource

Le contrôleur

Lorsque vous créez un contrôleur vous devez déterminer le nom des méthodes, les créer puis les coder. Un contrôleur de ressource peut bien vous aider dans cette tâche en normalisant (REST/CRUD) les méthodes. Pour créer un contrôleur de ressource c’est tout simple :

```
php artisan make:controller
UserController --resource
```

Vous avez alors la création du contrôleur comme on l’a déjà vu dans ce cours :



données

M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

Mais le fait d'ajouter l'option **--resource** crée ce code :

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class UserController extends Controller
{
    /**
     * Display a listing of the
     resource.
     *
     * @return
     \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a
     new resource.
     *
     * @return
     \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created
     resource in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     * @return
     \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified
     resource.
     *
     * @param int $id
     * @return
     \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }
}
```

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données



devotion1 dans Créer un blog – les

```

/**
 * Show the form for editing the
 * specified resource.
 *
 * @param int $id
 * @return
 * \Illuminate\Http\Response
 */
public function edit($id)
{
    //
}

/**
 * Update the specified resource
 * in storage.
 *
 * @param
 * \Illuminate\Http\Request $request
 * @param int $id
 * @return
 * \Illuminate\Http\Response
 */
public function update(Request
$request, $id)
{
    //
}

/**
 * Remove the specified resource
 * from storage.
 *
 * @param int $id
 * @return
 * \Illuminate\Http\Response
 */
public function destroy($id)
{
    //
}
}

```

On se retrouve avec 7 méthodes correspondant aux tâches classiques :

- **index** : afficher la liste des utilisateurs
- **create** : afficher le formulaire pour la création d'un utilisateur
- **store** : créer un utilisateur
- **show** : afficher l'utilisateur
- **edit** : afficher le formulaire pour la modification d'un utilisateur
- **update** : modifier un utilisateur
- **destroy** : supprimer un utilisateur

Évidemment il vous reste à coder toutes ces méthodes...

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

Les routes

Pour générer les routes vers les 7 méthodes du contrôleur de ressource c'est aussi très simple. Entrez ce code dans le fichier **routes/web.php** :

```
Route::resource('users', 'UserController');
```

Si maintenant vous regardez les routes générées avec :

```
php artisan route:list
```

Vous obtenez ceci :

GET HEAD	users	users.index	App\Http\Controllers\UserController@index
POST	users	users.store	App\Http\Controllers\UserController@store
GET HEAD	users/create	users.create	App\Http\Controllers\UserController@create
GET HEAD	users/{user}	users.show	App\Http\Controllers\UserController@show
PUT PATCH	users/{user}	users.update	App\Http\Controllers\UserController@update
DELETE	users/{user}	users.destroy	App\Http\Controllers\UserController@destroy
GET HEAD	users/{user}/edit	users.edit	App\Http\Controllers\UserController@edit

On a 7 routes, chacune correspondant à une méthode du contrôleur :

Verbe	URL	Méthode du contrôleur
GET	users	index()
GET	users/create	create()
POST	users	store()
GET	users/{user}	show()
GET	users/{user}/edit	edit()

données

M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...
Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)
Une petite vidéo pour se divertir sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)
Connexion

Liveware Fullcalendar
Flux des publications

Livewire Datatable
Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois
COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page d'accueil

noubis dans Cours Laravel 8 – les bases – la validation

devotion1 dans Créer un blog – les données

bestmomo dans Créer un blog – les données

devotion1 dans Créer un blog – les données

https://laravel.sillo.org/cours-laravel-5-5-les-donnees-les-ressources/

6/25

PUT/PATCH	users/{user}	update()
DELETE	users/{user}	destroy()

Un peu de code

Liste des utilisateurs

Maintenant qu'on a un contrôleur et des routes dédiées il suffit de compléter le code. On va commencer par déclarer l'espace de nom pour le modèle pour simplifier la syntaxe :

```
use App\User;
```

Ensuite on code la méthode **index** :

```
public function index()
{
    $users = User::all();
    foreach ($users as $user) {
        echo $user->name . '<br>';
    }
}
```

Et pour l'url **.../users** on obtient quelque chose dans ce genre :

```
Rigoberto Balistreri Sr.
Noble Fisher Sr.
Brenna Mohr
Helen Graham
Gardner Trantow
Adrianna Kautzer
Eladio Douglas IV
Susie Koch
Mr. Ned Harvey V
Kareem Hirthe
```

Afficher un utilisateur

C'est la méthode **show** qui est concernée :

```
public function show(User $user)
{
    echo 'Nom : ' . $user->name .
    '<br>';
}
```

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```
echo 'Email : ' . $user->email .
'<br>';
}
```

On a déjà vu dans un précédent chapitre la liaison implicite entre le paramètre dans l'url et un modèle.

Avec une url du genre **...users/1** on obtient :

```
Nom :Rigoberto Balistreri Sr.
Email :xstroman@example.com
```

Créer un utilisateur

Pour la création d'un utilisateur deux méthodes sont concernées :

```
public function create()
{
    return view('create');
}

public function store(Request
$request)
{
    User::create($request->all());

    return "Utilisateur créé !";
}
```

Pour simplifier je n'ai pas prévu de validation.

Voici la vue **create** :

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-
offset-2">
            <div class="panel panel-
default">
                <div class="panel-
heading">Créer un utilisateur</div>

                <div class="panel-
body">
                    <form
class="form-horizontal" method="POST"
action="{{ route('users.store') }}">
                        {{
csrf_field() }}

                    <div
class="form-group">
                        <label
for="name" class="col-md-4 control-
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les


```
<label">Nom</label>

                                <div
class="col-md-6">

<input id="name" type="text"
class="form-control" name="name"
required autofocus>

                                </div>
                            </div>

                                <div
class="form-group">

                                    <label
for="email" class="col-md-4 control-
label">E-Mail</label>

                                <div
class="col-md-6">

<input id="email" type="email"
class="form-control" name="email"
required>

                                </div>
                            </div>

                                <div
class="form-group">

                                    <label
for="password" class="col-md-4
control-label">Mot de passe</label>

                                <div
class="col-md-6">

<input id="password" type="password"
class="form-control" name="password"
required>

                                </div>
                            </div>

                                <div
class="form-group">

                                    <div
class="col-md-8 col-md-offset-4">

<button type="submit" class="btn btn-
primary">

Enregistrer

</button>

                                    </div>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
endsection
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

Un site de réservation avec la TALL

 CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

```
stack (partie 1)
```

Connexion

Liveware Fullcalendar

Flux des publications

..... Livewire Datable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois
COMMENTAIRES RÉCENTS

azizoux dans Créer un blog - la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

Avec cet aspect (en utilisant le template par défaut de Laravel) en utilisant l'url

.../users/create :

Remarquez dans le formulaire l'utilisation du verbe **POST** et la génération de l'action à partir du nom de la route :

```
<form class="form-horizontal"
method="POST" action="{{
route('users.store') }}">
```

Modifier un utilisateur

Pour la modification d'un utilisateur ce sont ces deux méthodes :

```
public function edit(User $user)
{
    return view('edit',
compact('user'));
}

public function update(Request
$request, User $user)
{
    $user->update($request->all());

    return "Utilisateur modifié !";
}
```

Pour simplifier je n'ai pas prévu de validation.

Voici la vue **edit** :

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-
offset-2">
            <div class="panel panel-
default">
                <div class="panel-
heading">Modifier un
utilisateur</div>
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !
ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois
COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

<div class="panel-body">
    <form
class="form-horizontal" method="POST"
action="{{ route('users.update',
$user->id) }}">
        {{
csrf_field() }}
        {{
method_field('PUT') }}

    <div
class="form-group">
        <label
for="name" class="col-md-4 control-label">Nom</label>
        <div
class="col-md-6">
            <input id="name" type="text"
class="form-control" name="name"
value="{{ $user->name }}" required
autofocus>
        </div>
    </div>

    <div
class="form-group">
        <label
for="email" class="col-md-4 control-label">E-Mail</label>
        <div
class="col-md-6">
            <input id="email" type="email"
class="form-control" name="email"
value="{{ $user->email }}" required>
        </div>
    </div>

    <div
class="form-group">
        <div
class="col-md-8 col-md-offset-4">
            <button type="submit" class="btn btn-primary">
Enregistrer
            </button>
        </div>
    </div>
</div>
@endsection

```

données, M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

Un site de réservation avec la TALL

 CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

```
stack (partie 1)
```

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datable

Flux des commentaires

Site de WordPress-FR

— ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog - la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog - les

données

bestmomo dans Créer un blog - les

données

devotion1 dans Créer un blog – les

On a du code commun avec la vue pour la création et dans un projet réel il serait judicieux de mutualiser ce code dans un template.

Avec cet aspect (en utilisant le template par défaut de Laravel) en utilisant l'url

.../users/1/edit :

Remarquez dans le formulaire la génération de l'action à partir du nom de la route (avec transmission de l'identifiant de l'utilisateur en paramètre) :

```
<form class="form-horizontal"
method="POST" action="{{
route('users.update', $user->id) }}">
```

D'autre part, comme on l'a déjà vu dans un précédent chapitre, le navigateur ne sait pas gérer le verbe PUT, donc on lui indique un verbe POST mais on ajoute un champ caché en précisant le bon verbe :

```
{{ method_field('PUT') }}
```

Supprimer un utilisateur

C'est la méthode **destroy** qui est concernée :

```
public function destroy(User $user)
{
    $user->delete();

    return 'Utilisateur supprimé !';
}
```

Avec une url du genre **...users/1** et le verbe **DELETE** on supprime l'utilisateur correspondant.

Pour voir ça en action on va ajouter une route :

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```
Route::get('users/{user}/destroy',
'UserController@destroyForm');
```

Une méthode dans le contrôleur :

```
public function destroyForm(User
$user)
{
    return view('destroy',
compact('user'));
}
```

Et cette vue **destroy** :

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-
offset-2">
            <div class="panel panel-
default">
                <div class="panel-
heading">Supprimer un
utilisateur</div>

                <div class="panel-
body">
                    <form
class="form-horizontal" method="POST"
action="{{ route('users.destroy',
$user->id) }}">
                        {{
csrf_field() }}
                        {{
method_field('DELETE') }}
                    <div
class="form-group">
                        <div
class="col-md-8 col-md-offset-4">
                            <button type="submit" class="btn btn-
primary">

Supprimer l'utilisateur {{ $user-
>name }}

                        </button>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection
```

Avec cet aspect pour une url dans le genre ...
users/1/destroy :

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

L'application d'exemple

Dans l'application d'exemple on utilise évidemment des ressources comme on l'a vu ci-dessus mais pas toujours de façon complète. Par exemple pour les contacts on utilise seulement (**only**) les méthodes **create** et **store** pour le front :

```
Route::resource('contacts',
'Front\ContactController', ['only' =>
['create', 'store']]);
```

Et les méthodes **index** et **destroy** pour le back :

```
Route::resource('contacts',
'ContactController', ['only' =>
['index', 'destroy']]);
```

Pour les commentaires on utilise seulement (**only**) les méthodes **update** et **destroy** pour le front :

```
Route::resource('comments',
'Front\CommentController', ['only' =>
['update', 'destroy']]);
```

Et les méthodes **index** et **destroy** pour le back :

```
Route::resource('comments',
'CommentController', ['only' =>
['index', 'destroy']]);
```

Pour les utilisateur dans le back on utilise seulement (**only**) les méthodes **index**, **edit**, **update** et **destroy** :

```
Route::resource('users',
'UserController', ['only' =>
['index', 'edit', 'update',
'destroy']]);
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

On pourrait adopter la logique inverse et utiliser **except** au lieu de **only**.

Par contre pour les articles en back la ressource est complète :

```
Route::resource('posts',
    'PostController');
```

Classiquement avec la méthode **index** on affiche la liste paginée de la ressource avec des boutons permettant les opérations de modification ou suppression. Par exemple dans l'administration pour les utilisateurs on a :

<

Pour chaque utilisateur on affiche les données et on a deux boutons à droite pour la modification et la suppression. Le premier est un simple lien du genre **...admin/users/1/edit** alors que le second est traité en Ajax.

Pour les articles on affiche 3 boutons :

On a en plus le bouton (vert) pour visualiser l'article.

Les API REST

REST (Representational State Transfer) est un style d'architecture pour constituer des API. Pour Laravel une API REST est :

- organisée autour de ressources

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

- sans persistance (aucune session)
- orientée client-serveur
- pouvant être mis en cache (donc une requête doit retourner toujours les mêmes résultats)
- retourner du JSON...

Si vous n'êtes pas trop sûr de vos connaissances concernant les API REST je vous conseille [ce cours](#).

Pour continuer avec notre ressource d'utilisateurs on peut imaginer ce **endpoint** pour les informations de tous les utilisateurs :

```
GET /api/users
[
  {
    id: 1,
    name: 'Rigoberto Balistreri
Sr. Noble Fisher Sr.'
  },
  {
    id: 2,
    name: 'Noble Fisher Sr.'
  }
  ...
]
```

Ou ce **endpoint** pour les informations concernant un utilisateur :

```
GET /api/users/2
{
  id: 2,
  name: 'Noble Fisher Sr.'
}
```

On peut imaginer d'autres endpoints pour modifier, ajouter, supprimer...

Il ne reste plus qu'à voir comment on réalise ça avec Laravel mais vous en avez désormais une idée déjà assez précise avec ce qu'on a déjà vu dans ce chapitre.

Le contrôleur

On va créer le contrôleur :

```
php artisan make:controller
Api/UserController --resource
```

données **M'OFFRIR UN CAFÉ**

[Si vous appréciez mon travail...](#)

[Merci à tous les généreux donateurs !](#)

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

[Une petite vidéo pour se divertir](#)

[sereinement !](#)

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

On le retrouve rangé dans le dossier

app/Http/Controllers/Api créé par la même occasion :

Avec évidemment le même code que celui qu'on a eu pour la même commande ci-dessus (et Laravel se débrouille tout seul pour les espaces de nom) :

```
<?php

namespace App\Http\Controllers\Api;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class UserController extends Controller
{
    /**
     * Display a listing of the
     resource.
     *
     * @return
     \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a
     new resource.
     *
     * @return
     \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created
     resource in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     * @return
     \Illuminate\Http\Response
     */
    public function store(Request
    $request)
    {
```

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

        //
    }

    /**
     * Display the specified
    resource.
     *
     * @param int $id
     * @return
    \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the
    specified resource.
     *
     * @param int $id
     * @return
    \Illuminate\Http\Response
     */
    public function edit($id)
    {
        //
    }

    /**
     * Update the specified resource
    in storage.
     *
     * @param
    \Illuminate\Http\Request $request
     * @param int $id
     * @return
    \Illuminate\Http\Response
     */
    public function update(Request
    $request, $id)
    {
        //
    }

    /**
     * Remove the specified resource
    from storage.
     *
     * @param int $id
     * @return
    \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        //
    }
}

```

On va supprimer les méthodes **create** et **edit** qui n'ont pas lieu d'être pour une API.

Et on va coder les autres méthodes :

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

<?php

namespace App\Http\Controllers\Api;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\User;

class UserController extends Controller
{
    /**
     * Display a listing of the
     resource.
     *
     * @return
     \Illuminate\Http\Response
     */
    public function index()
    {
        return User::all();
    }

    /**
     * Store a newly created
     resource in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     * @return
     \Illuminate\Http\Response
     */
    public function store(Request
    $request)
    {
        User::create($request->all());
    }

    /**
     * Display the specified
     resource.
     *
     * @param int $id
     * @return
     \Illuminate\Http\Response
     */
    public function show(User $user)
    {
        return $user;
    }

    /**
     * Update the specified resource
     in storage.
     *
     * @param
     \Illuminate\Http\Request $request
     * @param int $id
     * @return
     \Illuminate\Http\Response
     */
    public function update(Request
    $request, User $user)
    {

```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !
ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois
COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

        $user->update($request->all());
    }

    /**
     * Remove the specified resource
     * from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy(User $user)
    {
        $user->delete();
    }
}

```

Evidemment ça ressemble énormément à ce qu'on a fait précédemment et c'est normal puisqu'il s'agit du même genre de traitement sur les mêmes données.

Les routes

Pour les routes on adopte aussi une ressource dans le fichier **routes/api.php** :

```

Route::namespace('Api')->group(function() {
    Route::resource('users', 'UserController', ['except' => ['create', 'edit']]);
});

```

Ce qui donne ces routes :

Le fait de déclarer ces routes dans le fichier **routes/api.php** a pour effet de leur appliquer le middleware **api** au lieu de **web**. Si vous regardez dans le fichier **app/Http/Kernel.php** vous allez voir la différence que ça génère :

```

protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptC
        \Illuminate\Cookie\Middleware\AddQueue
        \Illuminate\Session\Middleware
        //
        \Illuminate\Session\Middleware\Authent
        \Illuminate\View\Middleware\ShareError
    ],
    'api' => [
        \Laravel\Sanctum\Http\Middleware\PreventFor
    ],
];

```

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

        \App\Http\Middleware\VerifyCs
\Illuminate\Routing\Middleware\Substit
    ],
    'api' => [
        'throttle:60,1',
        'bindings',
    ],
];

```

Comme on est sans état plus de cookie, de session, de protection CSRF...

Le fonctionnement

Maintenant avec l'url **...api/users** on obtient une réponse JSON (Laravel convertit automatiquement la réponse en JSON sans qu'on lui demande) :

On obtient toute les colonnes non cachées, c'est à dire celles qui ne sont pas définies dans la propriété **\$hidden** du modèle :

```

protected $hidden = [
    'password', 'remember_token',
];

```

Avec l'url **.../api/users/2** on obtient :

On pourrait aussi imaginer utiliser une pagination pour cette API. Il suffit de changer la méthode **index** :

```

public function index()
{
    return User::paginate(4);
}

```

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !
ARTICLES RÉCENTS

Un site de réservation avec la TALL
CLIN D'OEIL

stack (partie 3)
Une petite vidéo pour se divertir
sereinement !
Un site de réservation avec la TALL

stack (partie 2)
MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)
Connexion

Liveware Fullcalendar
Flux des publications

Livewire Datatable
Flux des commentaires

Site de WordPress-FR
ARCHIVES

Sélectionner un mois
COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page
d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

On peut alors demander la page qu'on veut, par exemple la seconde avec l'url **...api/users?page=2** :

On ne reçoit que les informations de la page 2 et en plus des renseignements concernant la pagination.

On peut imaginer aussi filtrer, ordonner...

Je ne traite pas dans ce chapitre de l'authentification spécifique aux API. Je vous invite à regarder [la documentation de Passport](#) sur le sujet.

Les ressources d'API

On a vu ci-dessus comme il est facile de retrouver une réponse JSON à partir des données d'un modèle. C'est parfait tant qu'on veut une simple sérialisation des valeurs des colonnes. Mais parfois on veut effectuer quelques traitements intermédiaires (ça correspond au design pattern transformer). Imaginons que nous voulons retourner seulement des deux premières lettres des noms, c'est une idée bizarre mais pourquoi pas ?

On utilise Artisan pour créer la ressource :

```
php artisan make:resource User
```

On se retrouve avec la ressource ici :

données **M'OFFRIR UN CAFÉ**

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

Avec ce code :

```
<?php

namespace App\Http\Resources;

use Illuminate\Http\Resources\Json\Resource

class User extends Resource
{
    /**
     * Transform the resource into an
    array.
     *
     * @param  \Illuminate\Http\Reque
     * @return array
     */
    public function toArray($request)
    {
        return
parent::toArray($request);
    }
}
```

Là on se contente de tout sérialiser, ça ne va donc pas changer grand chose à ce qu'on avait précédemment. Dans le contrôleur on utilise la ressource, par exemple pour la méthode **show** :

```
use App\Http\Resources\User as
UserResource;

...

public function show(User $user)
{
    return new UserResource($user);
}
```

Maintenant pour l'url **.../api/users/2** on obtient :

Maintenant changeons la ressource :

```
public function toArray($request)
{
```

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL

Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

```

return [
    'id' => $this->id,
    'name' => substr($this->name, 0, 2),
    'email' => $this->email,
    'created_at' => $this->created_at,
    'updated_at' => $this->updated_at,
];
}

```

Maintenant pour l'url **.../api/users/2** on obtient :

On voit qu'on n'a plus que les deux premiers caractères du nom. On a bien effectué un traitement intermédiaire sur les valeurs retournées.

Il y aurait encore beaucoup à dire sur les ressources et les API mais vous avez maintenant une bonne base de départ.

En résumé

- Un contrôleur de ressource permet de normaliser les opérations CRUD. Sa création et son routage sont très simples.
- Laravel permet de créer facilement une API et de permettre des transformations avec des ressources.

Partager :



Par *bestmomo*

2 Commentaires

2 COMMENTAIRES

données M'OFFRIR UN CAFÉ

Si vous appréciez mon travail...

Merci à tous les généreux donateurs !

ARTICLES RÉCENTS

Un site de réservation avec la TALL

CLIN D'OEIL

stack (partie 3)

Une petite vidéo pour se divertir

sereinement !

Un site de réservation avec la TALL

stack (partie 2)

MÉTA

Un site de réservation avec la TALL
Inscription

stack (partie 1)

Connexion

Liveware Fullcalendar

Flux des publications

Livewire Datatable

Flux des commentaires

Site de WordPress-FR

ARCHIVES

Sélectionner un mois

COMMENTAIRES RÉCENTS

azizoux dans Créer un blog – la page

d'accueil

noubis dans Cours Laravel 8 – les

bases – la validation

devotion1 dans Créer un blog – les

données

bestmomo dans Créer un blog – les

données

devotion1 dans Créer un blog – les

JAY

7 NOVEMBRE 2017 À 10 H 29 MIN

CONNECTEZ-VOUS POUR RÉPONDRE

Salut , je te laisse un commentaire sur ce fil concernant le chapitre LES ASSISTANT sur lequel je ne peux pas y acceder !!
je pense ya un probleme de droit.
peux tu corrigé cela ? merci

merci encore pour ton boulot !!

BESTMOMO

7 NOVEMBRE 2017 À 13 H 03 MIN

CONNECTEZ-VOUS POUR
RÉPONDRE

Salut,

Oui je n'avais pas mis le bon lien, c'est réparé,
merci 😊

données

ARTICLES RÉCENTS

Un site de réservation avec la TALL

stack (partie 3)

Un site de réservation avec la TALL

stack (partie 2)

Un site de réservation avec la TALL

stack (partie 1)

Liveware Fullcalendar

Livewire Datatable

ARCHIVES

Sélectionner un mois



LAISSER UN COMMENTAIRE

Vous devez **vous connecter** pour publier un commentaire.

Thème Ashe par WP Royal.

