



# Symfony se met au front à votre tour maintenant!



Merci PHP





Merci Laravel



# L'union fait la force

---



Il y a t'il des Js lovers ?



Je vous comprends

---

Je vous comprends

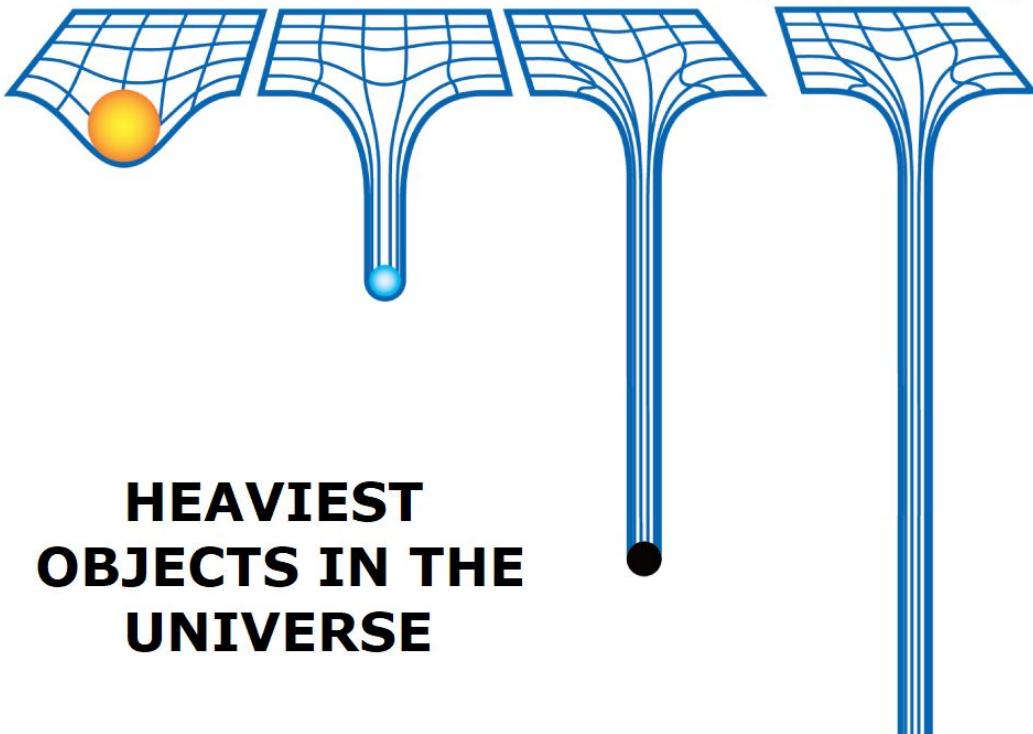


Sun

Neutron star

Black hole

`node_modules`



**HEAVIEST  
OBJECTS IN THE  
UNIVERSE**

# Les conséquences

---

Pov un dev back quand tu lui parles de front:



A vibrant illustration of a young blue elephant with large, expressive red eyes. It is pushing its way through a wall made of reddish-brown bricks. The elephant's trunk and front paws are visible as it breaks through the structure, with many bricks and stones flying off in all directions. The background is a bright, glowing yellow, suggesting either a sunset or a powerful energy source.

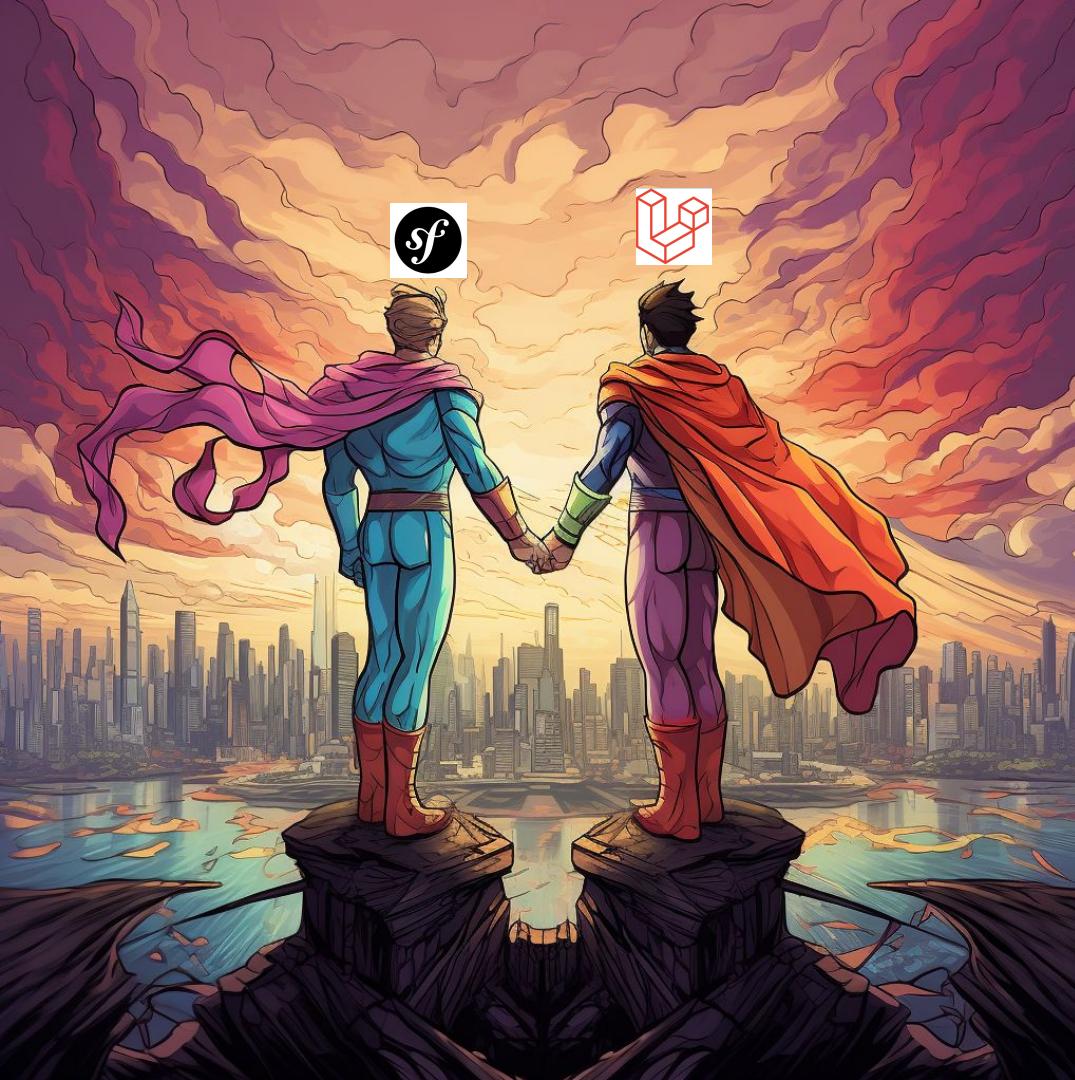
# Les conséquences

---



# Les conséquences

---



# Symfony et Laravel à notre secours

---

Système de build



Js

Un espace pour  
communiquer

# Symfony et Laravel à notre secours

# Attention

---



# Component Architecture



CoachLinkComponent

TrainingListComponent

PrimaryButton

# Les concepts essentielles

---

- Décomposition
- Indépendance
- Props
- State

# Anonymous

2 types

de composants

# Composant avec logique

# Anonymous Component

---

The screenshot shows the 'Bienvenue' (Welcome) screen of the Let's COACH app. At the top, there is a logo with a stylized bird icon and the text 'Let's COACH'. Below the logo, the word 'Bienvenue' is displayed in large blue letters, followed by the name 'Stef' in a smaller blue font. A yellow rectangular placeholder is shown below the name. The text 'Ici, tu peux retrouver tous les entraînements que t'as programmé. Savoir qui participe, et créer de nouvelles séances.' is present in a purple font. Underneath, there is a section titled 'Ton lien à partager:' with a URL 'https://www.li...tblin' and a button labeled 'COPIER LE LIEN' (Copy Link). The date 'MERCREDI 25 OCTOBRE' is prominently displayed. Below the date, there are two event cards. The first card for '09:10 ARTY SPEED' has the text 'Petit test de création d'une séance de coaching' and shows '10 PLACES RESTANTES'. The second card for another '09:10 ARTY SPEED' session also shows '10 PLACES RESTANTES'. At the bottom of the screen, a pink button labeled 'NOUVEL ENTRAINEMENT' (New Training) is highlighted with a green arrow pointing to it, labeled 'PrimaryButton'.

PrimaryButton



# Anonymous Component

```
{# components/Button/PrimaryButton #-}

{%- props label %}

<button class="btn btn-primary">
  {{ label }}
</button>
```

```
{# components/page/dashboard.html.twig #-}

{%- block content %}

  ...

  <twig:Button:PrimaryButton label="Nouvel entrainment"/>

{%- endblock %}
```

# Anonymous Component



```
{-- components/Button/PrimaryButton --
}
@props(['type' => 'label'])

<button class="btn btn-primary">
    {{ $label }}
</button>
```



```
{-- pages/Dashboard.blade --}

@section('content')
    <x-Button.PrimaryButton label="Nouvel entrainement"/>
@endsection
```

# Logical Component

---

The screenshot shows the Let's COACH mobile application interface. At the top, there is a logo with a stylized bird icon and the text "Let's COACH". Below the logo, the word "Bienvenue" is displayed, followed by the name "Stef". A yellow bar contains text about finding programmed trainings and creating new ones. Below this, a section titled "Ton lien à partager:" shows a URL and a "COPIER LE LIEN" button. The main content area is titled "MERCREDI 25 OCTOBRE". It lists two training sessions: "09:10 ARTY SPEED" with "10 PLACES RESTANTES" and "09:10 ARTY SPEED" with "10 PLACES RESTANTES". Both sessions have a "NOUVEL ENTRAINEMENT" button at the bottom.



TrainingListComponent

# Logical Component



```
//src/Components/TrainingList.php

namespace App\Components;

use Symfony\UX\TwigComponent\Attribute\AsTwigComponent;

#[AsTwigComponent]
class TrainingList
{
    public Coach $coach;

    public function __construct(
        private readonly TrainingRepository $trainingRepository
    ) {}

    public function getTrainings()
    {
        return $this->trainingRepository->findAllByCoach($this->coach);
    }
}
```

# Logical Component



```
{# components/TrainingList.html.twig #}

{% for training in this.getTrainings %}
    <twig:Card:TrainingCard training={{ training }}/>
{% endfor %}
```



```
{# pages/dashboard.html.twig #}

{% block content %}
    <twig:TrainingList coach={{ coach }}/>
{% endblock %}
```

# Logical Component



```
namespace App\View\Components;

use Illuminate\View\Component;
use Illuminate\View\View;

class TrainingList extends Component
{
    public function __construct(
        public Coach $coach,
        public TrainingRepository $trainingRepository
    ) {}

    public function getTrainingList(): array
    {
        return $this->trainingRepository->findAllByCoach($coach);
    }

    public function render(): View
    {
        return view('components.trainingList');
    }
}
```

# Logical Component



```
{{{-- components/TrainingList --}}}

@foreach ($trainings as $training)
    <x-card.trainingCard training="{{ training }}"/>
@endforeach
```



```
{{{-- pages/Dashboard --}}}

@section( 'content' )
    ...
    <x-trainingList coach="{{ $coach }}"/>
@endsection
```



# Live Component

```
#[AsLiveComponent]
class TrainingList
{
    use DefaultActionTrait;

    #[LiveProp]
    public int $pageIndex = 0;

    public function __construct(
        private readonly TrainingRepository $trainingRepository
    ) {}

    #[LiveAction]
    public function next()
    {
        $this->pageIndex++;
    }

    public function getTrainings(): array
    {
        return $this->trainingRepository->paginate(
            $this->coach,
            $this->pageIndex
        );
    }
}
```

# Live Component



```
<div {{ attributes }}>
  {% for training in this.getTrainings %}
    <twig:Card:TrainingCard training="{{ training }}"/>
  {% endfor %}

  <button
    data-action="live#action"
    data-action-name="next"
  >
    Voir plus
  </button>
</div>
```



# Live Component

```
namespace App\Http\Livewire;

use Livewire\Component;

class TrainingList extends Component
{
    public function __construct(
        public string $coach,
        public TrainingRepository $trainingRepository,
        public int $pageIndex = 0
    ) {}

    public function getTrainingList(): array
    {
        return $this->trainingRepository->paginateByCoach(
            $this->coach,
            $this->pageIndex
        );
    }

    public function next()
    {
        $this->pageIndex++;
    }

    public function render()
    {
        return view('components.trainingList');
    }
}
```

# Live Component

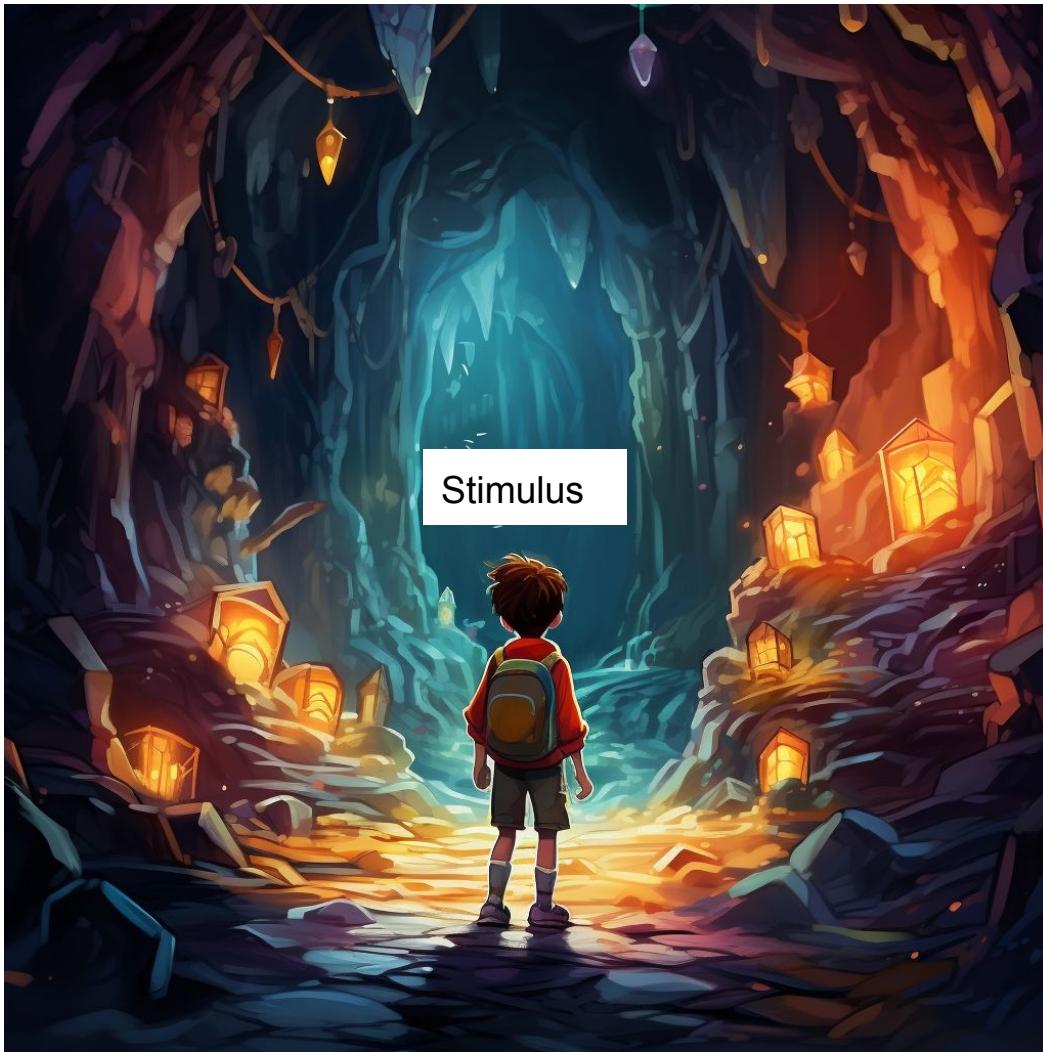
```
{{{-- components/TrainingList.blade --}}}

@foreach ($trainings as $trainings)
    <x-card.TrainingCard training="{{ $training }}"/>
@endforeach

<button wire:click="next">Voir plus</button>
```

# Stimulus

---





# Stimulus

```
<div data-controller="hello">
  <input data-hello-target="name" type=text>

  <button data-action="click->hello#greet">
    Greet
  </button>

  <span data-hello-target="output">
  </span>
</div>
```

```
// hello_controller.js

import { Controller } from "stimulus";

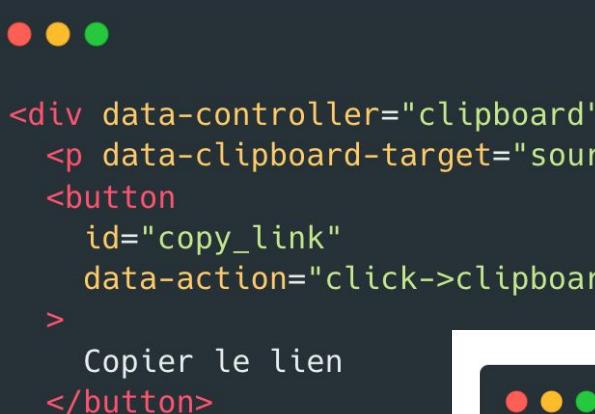
export default class extends Controller {
  static targets = ["name", "output"];

  greet() {
    this.outputTarget.textContent = `Hello, ${this.nameTarget.value}!`;
  }
}
```



# Stimulus

```
<div data-controller="clipboard">
  <p data-clipboard-target="source">{{ link }}</p>
  <button
    id="copy_link"
    data-action="click->clipboard#copy"
  >
    Copier le lien
  </button>
</div>
```



```
// clipboard_controller.js

import { Controller } from "@hotwired/stimulus";

export default class extends Controller {
  statics targets = ["source"];

  copy() {
    event.preventDefault();

    navigator.clipboard.writeText(this.sourceTarget.innerText);
  }
}
```



# Stimulus

```
{# components/TrainingList.html.twig #}

<div {{ attributes }}>
    {% for training in trainings %}
        <twig:Card:TrainingCard training={{ training }}/>
    {% endfor %}
    <button
        data-action="live#action"
        data-action-name="resetMax"
    >
        Voir plus
    </button>
</div>
```



```
<div
    data-controller="live"
    data-live-name-value="TrainingList"
    data-live-url-value="/_components/TrainingList"
    data-live-id="live-37010944-0"
    >
    ...
</div>
```

AutoComplete

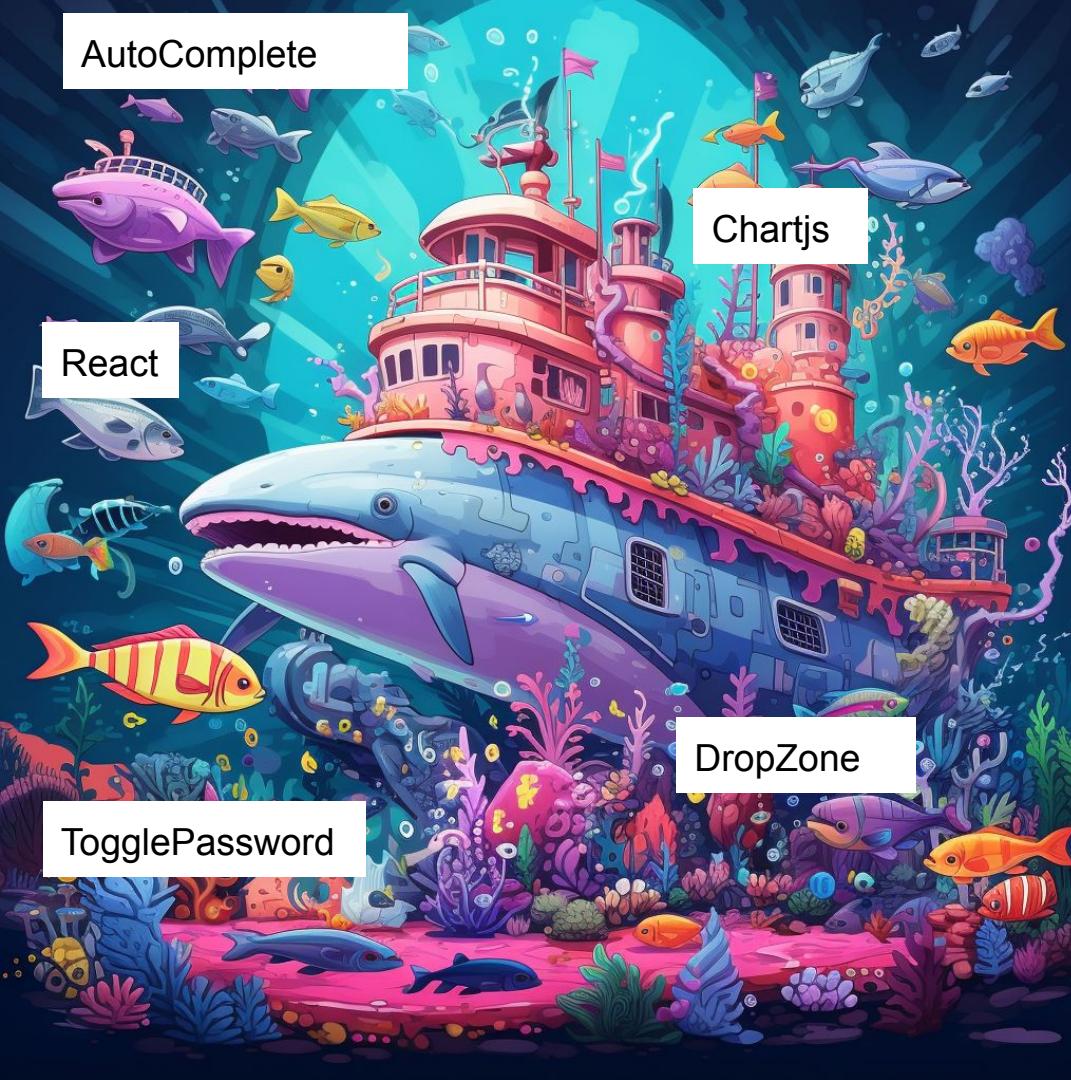
# L'écocystème

React

Chartjs

DropZone

TogglePassword



# L'écosystème

---



# Le build

---





# Le build

```
$ php bin/console importmap:require bootstrap
```



```
<script type="importmap">{
    "imports": {
        "app": "/assets/app-4e986c1a2318dd050b1d47db8d856278.js",
        "/assets/duck.js": "/assets/duck-1b7a64b3b3d31219c262cf72521a5267.js",
        "bootstrap": "https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/+esm"
    }
}</script>
```



```
import Alert from "bootstrap";

Alert.display("hello !");
```

# Amusez-vous bien

---



Mathéo Daninos

twitter: @matheo.daninos

github: webmamba



01

# Sommaire

01 -

02 -

03 -

04 -

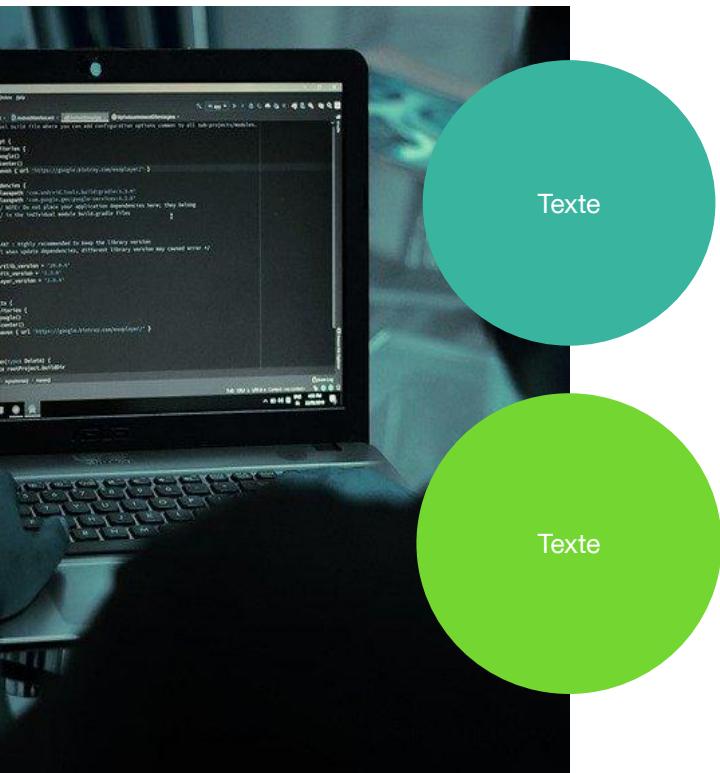
05 -

# Titre

---

# Titre

---



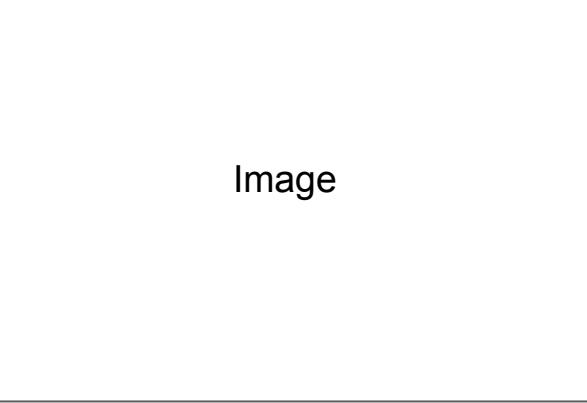
# Titre

---

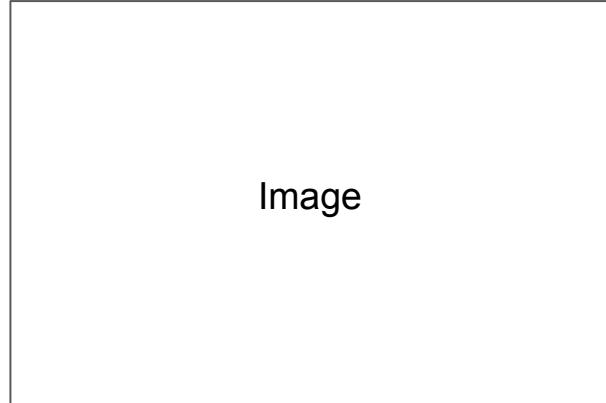
Image

# Titre

---



Image



Image

# Titre

---

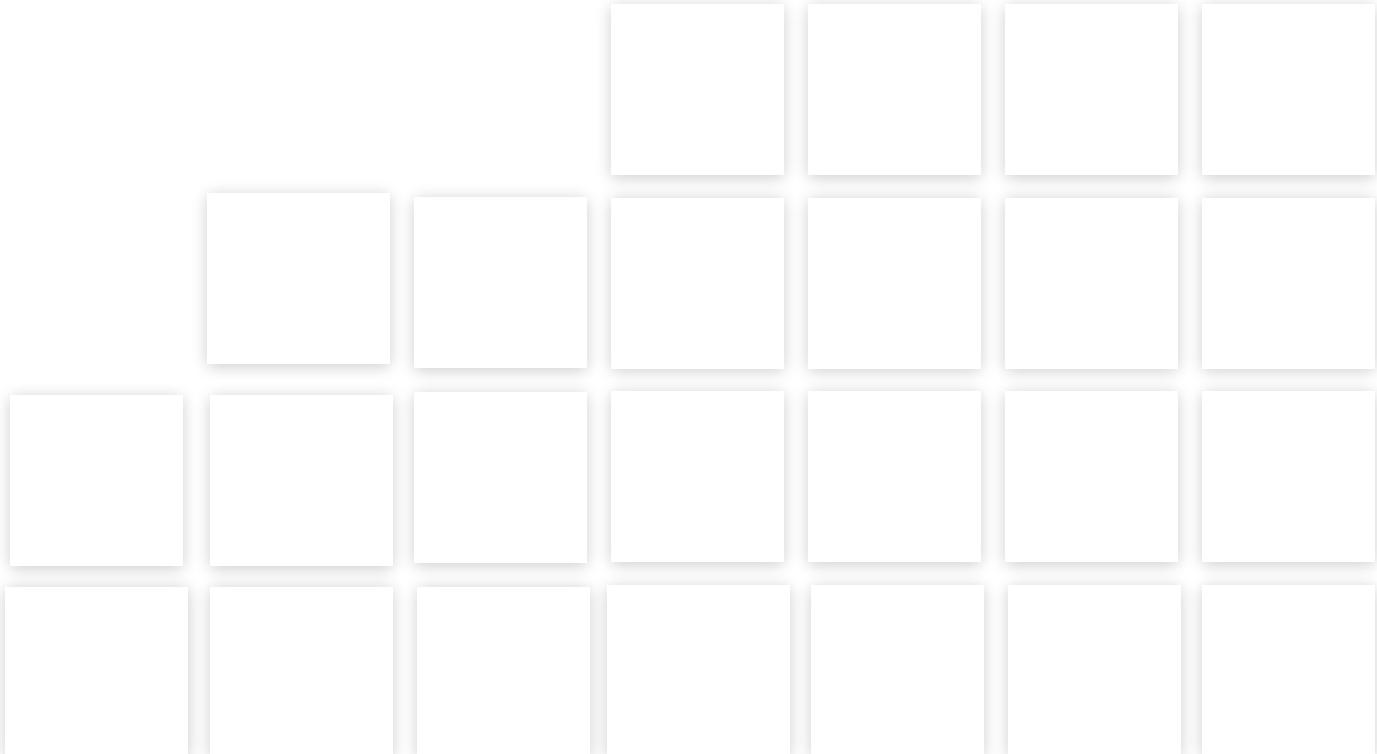


# Titre



# Titre

---



# Merci Laravel

- PhpLambo

# Partie 1

# Je vous comprend

tout les jours un nouveaux framework

- maintenance des packages
- difficile de recruter des équipes cohérentes
- peu de convention

# Je vous comprend

## Complexité inutile

- duplication de code avec le back
- nodes\_module
- configurer le système de build
- typescript

# Consequence

Fuites des developpeur Back

# Social

- Perte des jeunes
- Perte des femmes
- Equipe avec probleme de communication

# La reponse de Symfony et Laravel

Un ecocysteme familier

# La reponse de Symfony et Laravel

- creer un espace pour communiquer plus facilement avec le front (Component)
- systeme de build simplifier (AssetMapper)
- du code accessible pour tout le monde

# Attention

Symfony et Laravel ne veulent pas remplacer le Js

-

Transition pour la partie 2 - Voir comment le Js nous inspire

# Partie 2

# Les règles essentielles

- Décomposition
- Indépendant du contexte

# Les règles essentielles

- props et state
- props ne vont que dans un sens

# 2 types de composants

# AnonymousComponent

(Exemples de code Symfony)

- Composant sans logique
- props
- nouvelle syntax

# Anonymous Component

exemple de code pour Laravel

# Anonymous Component

Exemple d'utilisation pratique avec une interface

# TwigComponent

(montrer une interface contenant une liste d'utilisateur)

- composant contenant de la logique

# Twig Component

Example de code avec Symfony

# Laravel Component

Exemple de code avec Laravel

# Rajouter du dynamisme

(Exemple d'interface: tableau avec pagination)

# LiveComponent

exemple de code

# Livewire

Exemple de code

# Fonctionnement de Stimulus

(exemple basique de code)

- controller
- target
- les actions

# Fonctionnement de stimulus

Exemple du bouton copy to clipboard

# Fonctionnement des LiveComponent et Livewire

# Stimulus ne remplace pas React

Prendre l'exemple des composants React, Vue et Svelte

Montrer un exemple d'utilisation

# Turbo

- Turbo Drive
- comment le mettre en place

# Petit tour de l'ecosystem

Faire un petit tour des composant des composant Symfony et Laravel

# Build

- Presentation de AssetMapper

# ImportMap

Explique cette nouveauté importante pour le web

# Conclusion

Amusez vous

Besoin d'aide pour la contribution