

# Petite thérapie de couple je vous apprends à aimer le front-end



Encore!

---



API ?



Un peu de context



Une application

---

# Premier problème

---





Exception Logs 1 | Stack Trace

TwigError

## RuntimeError

► Show exception properties

- in templates/home.html.twig (line 4) □

```
1.  {% extends 'layout.html.twig' %}  
2.  
3.  {% block content %}  
4.      <p>{{ radio.name }}</p>  
5.      <twig:RadioList/>  
6.  {% endblock %}
```

+ in var/cache/dev/twig/3b/3b43f55b4ca496ffae57ad61ff803994.php -> **(closure)** (line 69)

+ in vendor/twig/twig/src/Template.php -> **block\_content** (line 171)

+ in var/cache/dev/twig/b3/b3e2f394a74f770b5aec16c9185fd185.php -> **displayBlock** (line 79)

+ in vendor/twig/twig/src/Template.php -> **block\_body** (line 171)

+ in var/cache/dev/twig/ec/ec83c9feb27c282cbcd2396924105a40.php -> **displayBlock** (line 71)

+ in vendor/twig/twig/src/Template.php -> **doDisplay** (line 394)

+ in vendor/twig/twig/src/Template.php -> **displayWithErrorHandling** (line 367)

+ in var/cache/dev/twig/b3/b3e2f394a74f770b5aec16c9185fd185.php -> **display** (line 49)

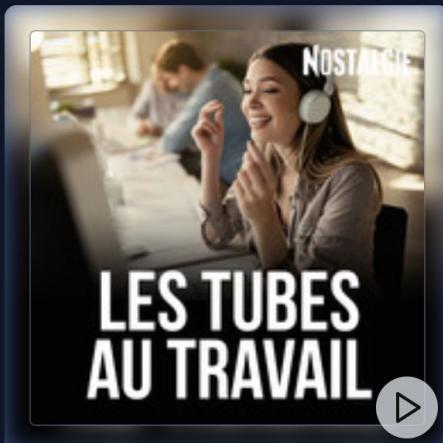


```
#[Route('/')]
public function index(): Response
{
    return $this->render('home.html.twig', [
        'radio' => $this->radioRepository->findOneBy(['name' =>
'Nostalgie'])
    ]
}
```



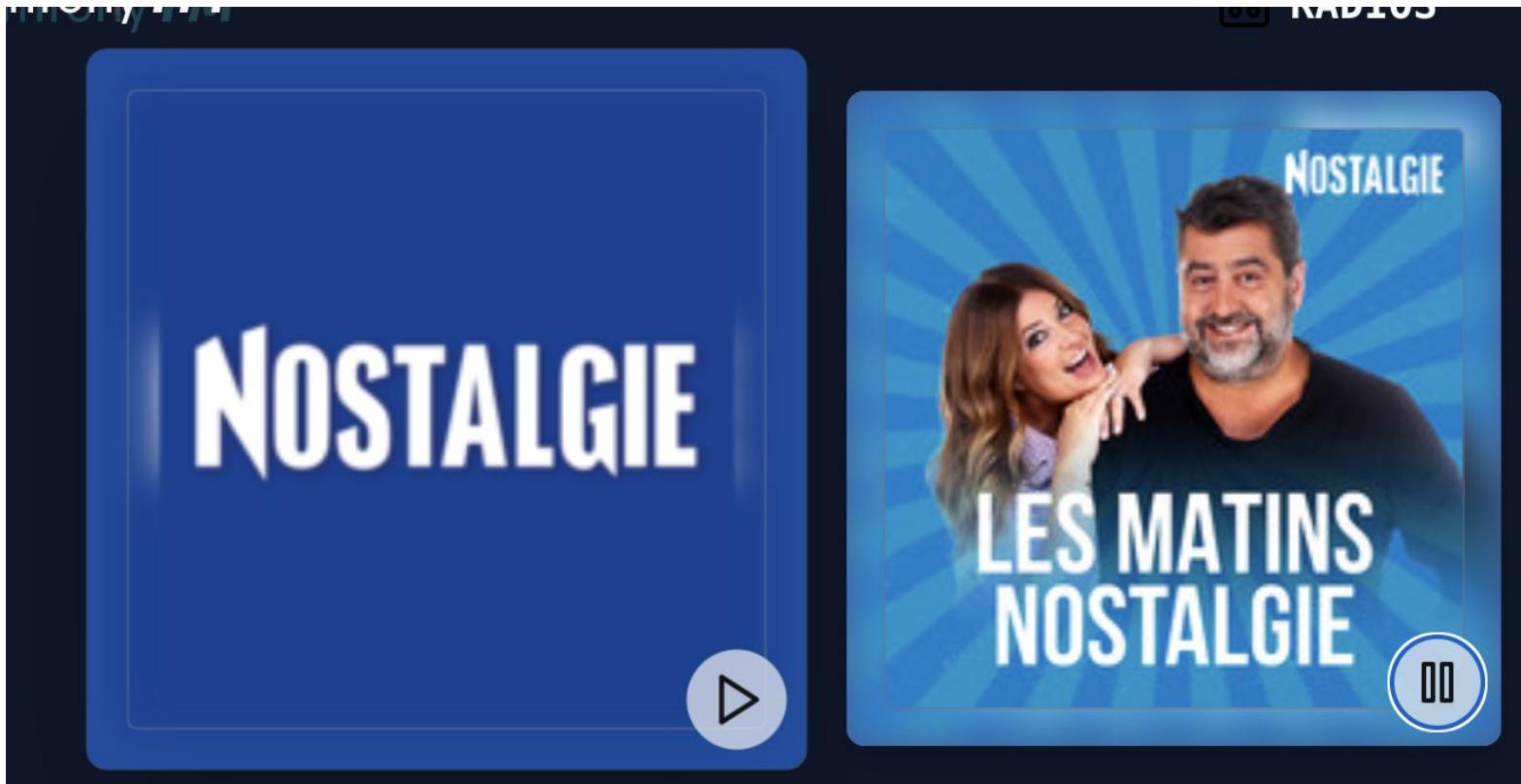
# Architecture component

---



## Props et States

---





```
#[AsTwigComponent]
final class RadioList
{
    public ?string $filter = null;

    public function __construct(
        private readonly RadioRepository $radioRepository
    ) {}

    public function getRadios(): array
    {
        if ($this->filter === null) {
            return $this->radioRepository->findAll();
        }

        return $this->radioRepository->findByGenre($this-
>filter);
    }
}
```

```
<div class="...">
    {%
        for radio in this.getRadios %}
        <div class="...">
            <img class="..."/>
            <div class="...">
                
            </div>
            <button class="...">
                <twig:UX:Icon name="pause"/>
                <twig:UX:Icon name="play"/>
            </button>
        </div>
    {%
        endfor %}
</div>
```



```
#[Route('/')]
public function index(): Response
{
    return $this->render('home.html.twig', [
        'radio' => $this->radioRepository->findOneBy(['name' =>
'Nostalgie;'])
    ]
}
```



```
{% extends 'layout.html.twig' %}

{% block content %}
    <twig:RadioList/>
{% endblock %}
```

# NOSTALGIE



RadioCard

Button

Icon

## FEMMES DES ANNEES 80

### LES TUBES AU TRAVAIL



### JOURNÉE INTERNATIONALE DU DISCO



### BONNE HUMEUR





Anonymous comasant

---

## /components/RadioCard.html.twig

```
● ● ●

<div
  class="
    transition ease-in-out delay-150 hover:-translate-y-1
    hover:scale-110 duration-300 aspect-square relative overflow-hidden
    rounded-lg bg-white/50 shadow-lg transition duration-300
    ease-in-out hover:shadow-2xl p-4"
  data-controller="radio"
  data-radio-source-value="{{ radio.url }}"
  data-radio-stream-value="{{ path('app_home_playerstream', {radio: radio.id}) }}"
  data-action="click->radio#toggle"
>
  {% block content %}
  {% endblock %}
</div>
```

## /components/RadioButton.html.twig



```
<button data-action="player#toggle" class="flex justify-center items-center position w-[2.5rem] h-[2.5rem] bg-slate-200/75 rounded-100 rounded-full">
    <div class="hidden" data-player-target="play" data-radio-target="play">
        <twig:UX:Icon name="pause" class="text-slate-50" style="color: black; opacity: .9; fill: none"/>
    </div>
    <div class="pause" data-player-target="pause" data-radio-target="pause">
        <twig:UX:Icon name="play" class="text-slate-50" style="color: black; opacity: .9; fill: none"/>
    </div>
</button>
```

```
{% props radio %}

<twig:RadioCard>
    <twig:RadioCardBackground
        logo="{{ radio.logo }}"
        radio="{{ radio.name }}"
        color="{{ radio.color }}"
    />
    <twig:RadioCardImage
        logo="{{ radio.logo }}"
        name="{{ radio.name }}"
    />
    <twig:RadioCardPlayer/>
</twig:RadioCard>
```



```
<div class="...">
    {% for radio in this.getRadios %}
        <div class="...">
            <img class="..."/>
            <div class="...">
                
            </div>
            <button class="...">
                <twig:UX:Icon name="pause"/>
                <twig:UX:Icon name="play"/>
            </button>
        </div>
    {% endfor %}
</div>
```



```
{% for radio in this.getRadios %}  
    <twig:Radio :radio="radio" />  
{% endfor %}
```

On a un problème Arthur !

RADIOS

GENRE

ANNÉES

Bonsoirreee Giselle !

CVA

# NOSTALGIE



Punaise John !



Sensio  
Labs

```
● ● ●

{%- props color, size, position %}

{%- set alert = cva({
    base: 'min-w-96 absolute rounded-lg shadow-md bg-slate-200 text-white',
    variants: {
        color: {
            'blue': 'bg-blue-800 shadow-blue-800',
            'red': 'bg-red-800 shadow-red-800',
            'green': 'bg-green-800 shadow-green-800',
        },
        size: {
            'sm': 'p-2 mb-2 text-sm',
            'lg': 'p-4 mb-4text-lg',
        },
        position: {
            'top-left': 'top-4 left-4',
            'top-right': 'top-4 right-4',
            'bottom-left': 'bottom-4 left-4',
            'bottom-right': 'bottom-4 right-4',
        }
    }
}) %}

<div class="{{ alert.apply({color, size, position}) }}" role="alert">
    {%- block content %}
    {%- endblock %}
</div>
```



```
<twig:Alert color='red' size="lg" position='bottom-left'>On un problème Arthur</twig:Alert>
<twig:Alert color='green' size="lg" position='bottom-left'>Nice to meet you</twig:Alert>
<twig:Alert color='blue' size="sm" position='top-left'>Bonjour Germaine</twig:Alert>
```

# Design System

---



# Storybook

---



# DEMO TIME

# SymfonyUI





Ajouter du dynamisme



Stimulus

---



```
<div
    class="..."
    data-controller="radio"
    data-radio-source-value="{{ radio.url }}"
    data-radio-stream-value="{{ path('app_home_playerstream', {radio: radio.id}) }}"
    data-action="click->radio#toggle"
>
    {% block content %}
    {% endblock %}
</div>
```



```
<div
    class="..."
    data-controller="radio"
    data-radio-source-value="{{ radio.url }}"
    data-radio-stream-value="{{ path('app_home_playerstream', {radio: radio.id}) }}"
    data-action="click->radio#toggle"
>
    {% block content %}
    {% endblock %}
</div>
```



```
<div
    class="..."
    data-controller="radio"
    data-radio-source-value="{{ radio.url }}"
    data-radio-stream-value="{{ path('app_home_playerstream', {radio: radio.id}) }}"
    data-action="click->radio#toggle"
>
    {% block content %}
    {% endblock %}
</div>
```



```
<div
    class="..."
    data-controller="radio"
    data-radio-source-value="{{ radio.url }}"
    data-radio-stream-value="{{ path('app_home_playerstream', {radio: radio.id}) }}"
    data-action="click->radio#toggle"
>
    {% block content %}
    {% endblock %}
</div>
```

```
export default class extends Controller {
    static values = {source: String, stream: String};

    static targets = ['audio', 'play', 'pause'];

    initialize() {
        this.play = false;
    }

    toggle(event) {
        event.stopImmediatePropagation();

        if (!this.play) {
            this.dispatch('playRadio', {
                detail: {
                    source: this.sourceValue,
                    stream: this.streamValue
                }
            });

            this.playTarget.classList.remove('hidden');
            this.pauseTarget.classList.add('hidden');
            this.play = true;

            return;
        }
    ...
}
}
```

```
export default class extends Controller {
    static values = {source: String, stream: String};

    static targets = ['audio', 'play', 'pause'];

    initialize() {
        this.play = false;
    }

    toggle(event) {
        event.stopImmediatePropagation();

        if (!this.play) {
            this.dispatch('playRadio', {
                detail: {
                    source: this.sourceValue,
                    stream: this.streamValue
                }
            });

            this.playTarget.classList.remove('hidden');
            this.pauseTarget.classList.add('hidden');
            this.play = true;

            return;
        }
    ...
}
}
```

```
export default class extends Controller {  
    static values = {source: String, stream: String};  
  
    static targets = ['audio', 'play', 'pause'];  
  
    initialize() {  
        this.play = false;  
    }  
  
    toggle(event) {  
        event.stopImmediatePropagation();  
  
        if (!this.play) {  
            this.dispatch('playRadio', {  
                detail: {  
                    source: this.sourceValue,  
                    stream: this.streamValue  
                }  
            });  
  
            this.playTarget.classList.remove('hidden');  
            this.pauseTarget.classList.add('hidden');  
            this.play = true;  
  
            return;  
        }  
  
        ...  
    }  
}
```



# LiveComponent

---





# TurboDrive

---

Turbo8

---



# DEMO TIME

A vibrant, stylized illustration of a car's rear end, focusing on the exhaust system. The car is dark blue with a metallic finish. Two large, multi-colored exhaust pipes (blue, green, yellow, red) exit the vehicle, emitting a massive, swirling plume of smoke that transitions through the spectrum of colors. The background features horizontal motion streaks in shades of blue, white, and orange, suggesting speed. The overall aesthetic is dynamic and energetic.

# TurboFrame

---

# DEMO TIME

```
{% extends 'base.html.twig' %}

{% block body %}
    <turbo-frame id="content">
        <header class="p-4 backdrop-blur-md">
            <twig:NavBar/>
        </header>
        <div class="container mx-auto px-4">
            {% block content %}
            {% endblock %}
        </div>
    </turbo-frame>
    <twig:RadioPlayer/>
    {% endblock %}
```



# TurboStream

# DEMO TIME



```
[Route('/player/{radio}')]
public function playerStream(Request $request, Radio $radio): Response
{
    $request->setRequestFormat(TurboBundle::STREAM_FORMAT);

    return $this->render('player.stream.html.twig', [
        'song' => $radio->getCurrentSong()
    ]);
}
```



```
<turbo-stream action="replace" target="player-radio">
  <template>
    <div class="flex items-center gap-2" id="player-radio">
      
        <p class="text-xl font-bold">{{ song.title }}</p>
        <p class="text-slate-400">{{ song.artist }}</p>
      </div>
    </div>
  </template>
</turbo-stream>
```

# Amusez vous !

---



Mathéo Daninos

twitter: @matheo.daninos

github: webmamba

