1.08

TEMPS 1 · LE PORTFOLIO · SÉANCE 2

Naviguer

Thibault Six · IFOSUP Wavre · 2026-2027



? Au clic sur Projets, qu'est-ce que le navigateur redemande ?

Avec <a>, le navigateur redemande tout ; avec <Link>, le JSON de la page.

`<a href="/projets">` **le navigateur redemande tout**

Nom	Type	Statut
projets	document	**200**
app.css	stylesheet	200
app.ts	script	200
Index.vue	script	200
… les scripts de Vue et du kit		200
quiz-capitales.jpg	img	200

tout est effacé puis redessiné, menu compris

`<Link href="/projets">` **le document reste en place**

Nom	Type	Statut	
{} **projets**	xhr	**200**	JSON
app.css	stylesheet	304	inchangée
Index.vue	script	200	1er passage
quiz-capitales.jpg	img	200	

l'en-tête et le menu restent, seul le centre change

6 lignes vermillon contre 1 ligne orange

`la réponse à projets, en-têtes dans l'onglet Réseau`

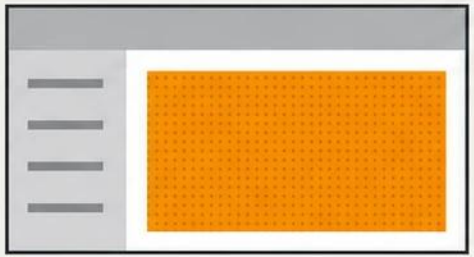
Content-Type	application/json
X-Inertia	true

Comptez les lignes vermillon : c'*est* ce que <Link> évite. Laravel exécute pourtant la même route.

? Que contient la réponse du serveur à un clic sur Link ?

La réponse à un clic sur <Link> est du JSON : le nom du composant et ses props, rien d'autre.

le navigateur



Vue remplace le composant de la page et met à jour l'adresse

clic sur <Link> `X-Inertia: true` →

le serveur Laravel

la même route, le même contrôleur

←

```
HTTP/1.1 200 OK
X-Inertia: true
Content-Type: application/json
Vary: X-Inertia

{"component":"projects\/Show",
"props":{"errors":{},"name":"Laravel",
"auth":{"user":null},"sidebarOpen":true,
"project":{"title":"Quiz des capitales",…}},…}
```

la feuille de style et les scripts déjà chargés restent en place : le passage d'une page à l'autre est plus rapide

l'onglet Réseau après un clic sur Projets : huit lignes, dont projets →

app.css	Index.vue?t=1793293043830	quiz-capitales.jpg	favicon.svg
projets	meteo-wavre.jpg	carnet-recettes.jpg	favicon.ico

Lisez la réponse de haut en bas : trois en-têtes, puis l'objet de page. Pas de HTML.

? Que se passe-t-il si l'adresse /projets change dans routes/web.php ?

Une adresse écrite deux fois, dans la route et dans la page, casse en silence : le lien mène à une 404.

routes/web.php

```
realisations
Route::get('projets/{project:slug}',
    [ProjectController::class, 'show'])
    ->name('projects.show');
```

la même adresse, écrite deux fois

resources/js/pages/projects/Index.vue

```
<a :href="`/projets/${project.slug}`">Voir le projet</a>
```

inchangé : l'adresse est écrite en dur

404 sans aucun message d'erreur au moment de la modification

L'adresse vit à deux endroits ; quand l'un change, rien ne prévient que l'autre est faux.

1.08

? Comment n'écrire l'adresse qu'une seule fois ?

Wayfinder lit vos routes PHP et écrit une fonction par route nommée, que vos pages importent.

routes/web.php

```
->name('projects.show')
```

composer run dev

```
vite | [vite+] (client) info: Types generated
for actions, routes, form variants
```

à chaque enregistrement de **routes/web.php**

resources/js/routes/projects/index.ts

```
export const show = (args: …): RouteDefinition<'get'> => ({
    url: '/projets/{project}',
```

projects/Index.vue

```
import { show } from '@/routes/projects';
<Link :href="show(project)">Voir le projet</Link>
```



générés dans resources/js, ignorés par git

@/ est le raccourci du kit pour resources/js/

Suivez la ligne : la route nommée, Vite qui génère, le fichier TypeScript, l'import dans la page. Renommez projets en realisations : chaque Link suit.

1.08

? Pourquoi show(project) accepte-t-il l'objet reçu en prop ?

show(project) lit la clé slug du projet et construit l'adresse /projets/quiz-des-capitales.

la prop `project`

```
{ "title": "Quiz des capitales",
  "slug": "quiz-des-capitales", … }
```

→ **`show(project)`** →

ce que `<Link>` sait lire

```
{ url: '/projets/quiz-des-capitales',
  method: 'get' }
```

- `string`
- `{ project: string | { slug: string } }`
- `[project: …]`
- `{ slug: string }` — **un objet qui possède une clé slug : votre prop en a une**

`url: '/projets/{project}'` dans `show.definition`

```
<a href="/projets/quiz-des-capitales">
```

Suivez le slug : dans la prop, dans le type accepté, dans l'adresse construite.
La page n'écrit jamais l'adresse.

1.08

? Comment donner à toutes les pages publiques le même en-tête et le même pied de page ?

Un layout entoure la page ; <slot /> marque l'endroit où elle s'affiche.

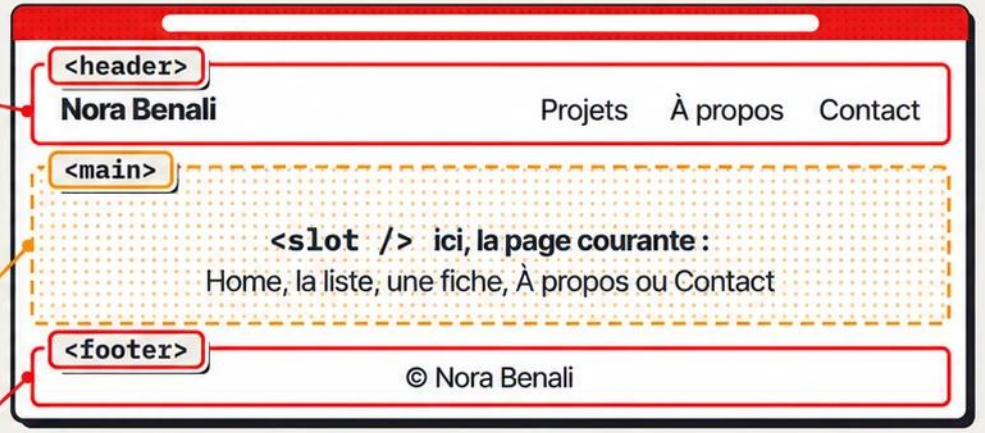
resources/js/layouts/PublicLayout.vue

```
<header>
    <Link :href="home()">Nora Benali</Link>
    <nav>
        <Link :href="projectsIndex()">Projets</Link>
        <Link :href="about()">À propos</Link>
        <Link :href="contact()">Contact</Link>
    </nav>
</header>

<main>
    <slot />
</main>

<footer>© Nora Benali</footer>
```

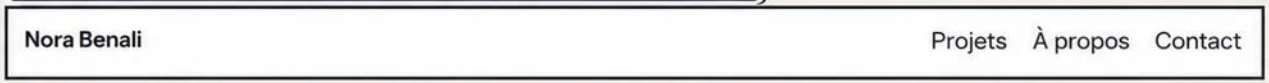
extrait : imports et classes Tailwind omis



d'une page à l'autre, l'en-tête et le pied restent en place ;
seul le contenu du slot change

`index as projectsIndex` : la fonction est renommée à l'import pour éviter la collision entre fichiers de routes

`l'en-tête du layout public, sur la fiche du quiz`



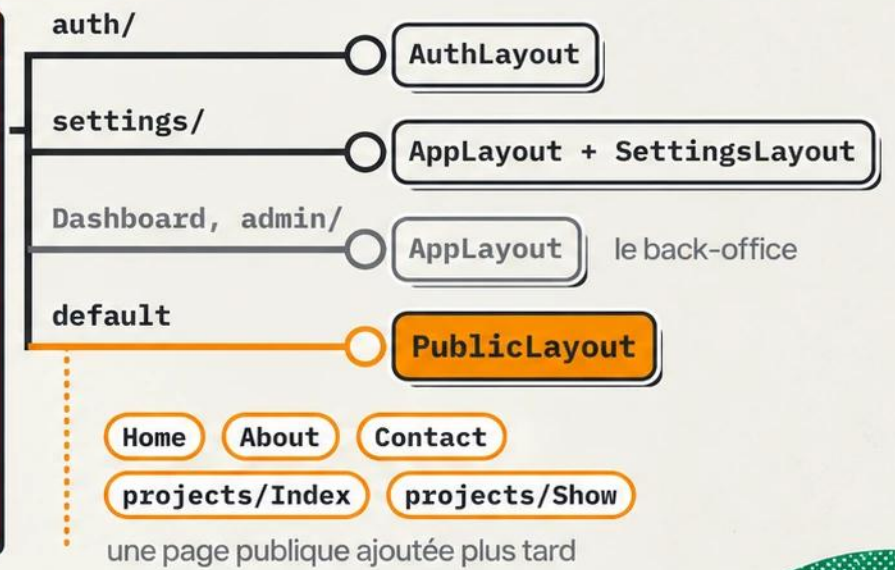
Lisez l'écran de haut en bas : l'en-tête, la zone de la page, le pied. Chaque bande est une balise du code ; au clic, seule la bande du milieu change.

? Qui décide quelle page reçoit PublicLayout ?

Le cas default du résolveur de app.ts donne PublicLayout à toutes les pages publiques, présentes et futures.

resources/js/app.ts

```
layout: (name) => {
    switch (true) {
        case name.startsWith('auth/'):
            return AuthLayout;
        case name.startsWith('settings/'):
            return [AppLayout, SettingsLayout];
        case name === 'Dashboard' || name.startsWith('admin/'):
            return AppLayout;
        default:
            return PublicLayout;
    }
},
```



l'autre écriture, `defineOptions({ layout: PublicLayout })`, se répète dans chaque page ; une page qui l'oublie s'affiche sans menu

Lisez les case dans l'ordre : switch (true) s'arrête au premier vrai ; tout ce qui n'est ni connexion, ni réglages, ni administration finit dans default.

1.08

? D'où vient le titre de l'onglet ?

<Head title> donne son titre à chaque onglet ; le suffixe vient de APP_NAME, dans .env.

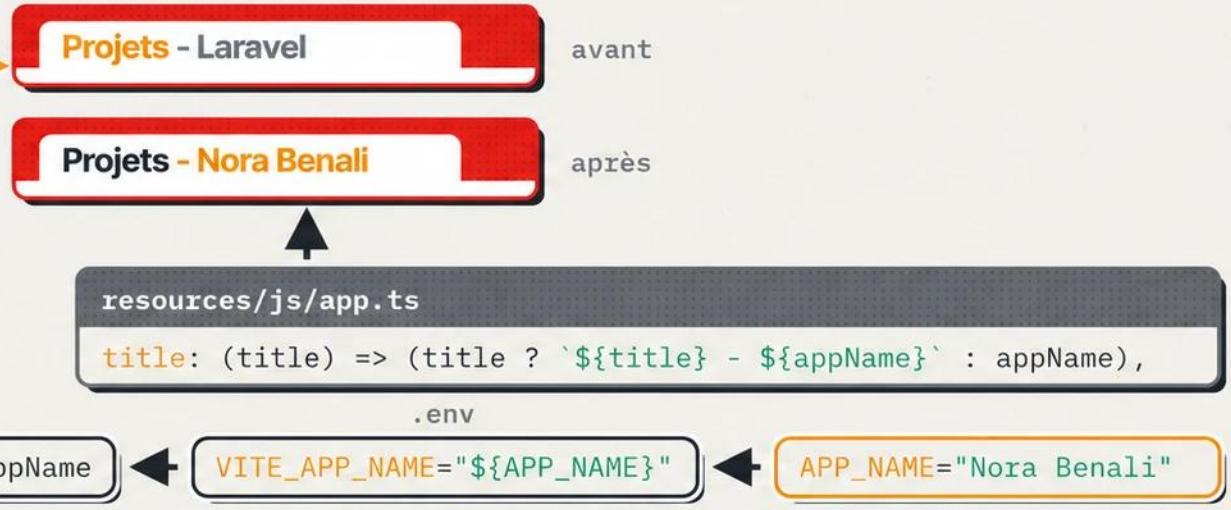
```
import { Head } from '@inertiajs/vue3';
```

```
<Head title="Projets" />
```

projects/Index.vue

```
<Head :title="project.title" />
```

projects/Show.vue · calculé à partir de la prop



remplacez Laravel par votre nom dès maintenant, dans `.env` ; relancez `composer run dev` si l'onglet ne change pas

Home.vue : **Accueil** · About.vue : **À propos** · Contact.vue : **Contact** · projects/Index.vue : **Projets**

Deux sources pour un titre : la page donne le début, `app.ts` ajoute le suffixe lu dans `.env`. Changez APP_NAME aujourd'hui, pas au chapitre 1.19.

? Une adresse inconnue, que voit le visiteur, et pourquoi s'en occuper ?

Une page 404 à vous : le visiteur garde votre menu et un lien pour repartir, le statut reste 404.

POURQUOI

404 | Not Found

la page du framework : ni votre menu, ni aucun lien, le visiteur repart

- ✓ le visiteur reste dans votre site
- ✓ un message en français et un lien pour repartir
- ✓ le statut HTTP reste 404, comme avant

Nora Benali Projets · À propos · Contact

404 · Page introuvable
Cette page n'existe pas ou n'est plus en ligne.
Retour à l'accueil

votre page, dans votre site

COMMENT, EN DEUX FICHIERS

1 app/Providers/AppServiceProvider.php

```php
Inertia::handleExceptionsUsing(function (ExceptionResponse $response) {
    if (in_array($response->statusCode(), [403, 404])) {
        return $response->render('ErrorPage', [
            'status' => $response->statusCode(),
        ])->withSharedData();
    }

    return null;
});
```

pour une 404 ou une 403, Laravel rend votre page `ErrorPage` avec le code en prop ; `return null` : les autres erreurs ne changent pas

```
'status' => 404
```

2 resources/js/pages/ErrorPage.vue

```
defineProps({ status: Number })
computed() · titre et message selon status
<Link :href="home()">Retour à l'accueil</Link>
```

le layout public vient du cas `default` du résolveur : le menu est là

Lisez d'abord le haut : ce que le visiteur gagne. Puis les deux fichiers : le premier attrape l'erreur et choisit votre page, le second l'affiche, avec le menu.

? Qu'est-ce qui doit être vrai à la fin ?

À vous, vingt-cinq minutes : toutes les pages publiques se parcourent depuis le menu, sans jamais taper d'adresse.

À FAIRE

1. Les routes `about` et `contact`, et les pages `About.vue`, avec le texte de votre devoir, et `Contact.vue`, vide pour l'instant.
2. `PublicLayout.vue` avec votre nom et le menu Projets, À propos, Contact, tous en `<Link>` vers des fonctions de Wayfinder.
3. Le résolveur de `app.ts` qui donne `PublicLayout` par défaut.
4. Dans la liste, le lien vers la fiche en `<Link :href="show(project)">`.
5. Un `<Head>` dans chaque page.
6. `Inertia::handleExceptionsUsing()` et `ErrorPage.vue`.

À VÉRIFIER

25 min

- [ ] Depuis l'accueil, vous atteignez les projets, une fiche, l'à-propos et le contact, puis vous revenez à l'accueil par votre nom, sans jamais taper d'adresse.
- [ ] Dans l'onglet Réseau, un clic sur « Projets » ne recharge pas la page : la requête `projets` est de type `xhr`, avec l'en-tête `X-Inertia`, et aucun document HTML n'est redemandé.
- [ ] Chaque onglet porte le titre de sa page.
- [ ] `/projets/inconnu` et `/nimporte-quoi` affichent votre page 404, dans votre layout.
- [ ] Votre code Vue ne contient aucune adresse écrite en dur.

Six étapes à gauche, cinq vérifications à droite ; la dernière se vérifie en cherchant « `/projets` » dans vos fichiers Vue.