1.13

TEMPS 1 · LE PORTFOLIO · SÉANCE 3

Modifier et supprimer

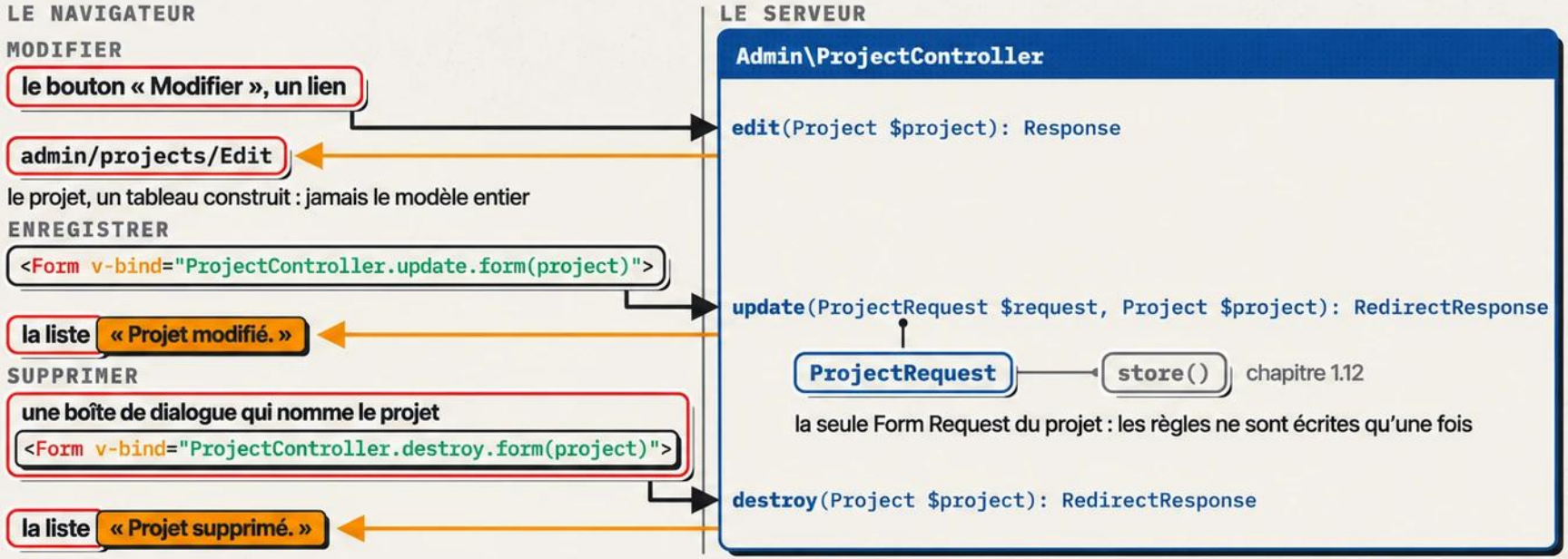
Thibault Six · IFOSUP Wavre · 2026-2027

5XDYN · Projet web dynamique · 1.13



? Que faut-il ajouter au back-office, côté navigateur et côté serveur, pour modifier et supprimer ?

Chaque bouton envoie sa requête, et le contrôleur y répond par edit(), update() ou destroy().



Lisez chaque rangée de gauche à droite, puis le retour : le bouton, la méthode du contrôleur, la page ou la liste qui revient.

? Le chapitre ne donne pas le code. Où trouver un modèle pour chaque morceau à écrire ?

Presque tout existe déjà : chaque morceau à écrire a son modèle, dans votre projet ou dans le kit.

POUR	PARTEZ DE	
la page de modification	`Create.vue`	chapitre 1.12 — même formulaire, autre fonction de Wayfinder, valeurs de départ
des champs préremplis	`resources/js/pages/settings/Profile.vue`	préremplit le nom et l'adresse e-mail
une suppression confirmée	`resources/js/components/DeleteUser.vue`	un Dialog qui contient un `<Form>`
la logique du serveur	`store()`	chapitre 1.12 — **5XCOS, chapitre 16** — « Formulaires : modifier et supprimer »
le bouton-lien « Modifier »	le bouton « Nouveau projet »	chapitre 1.11 — avec `as-child`

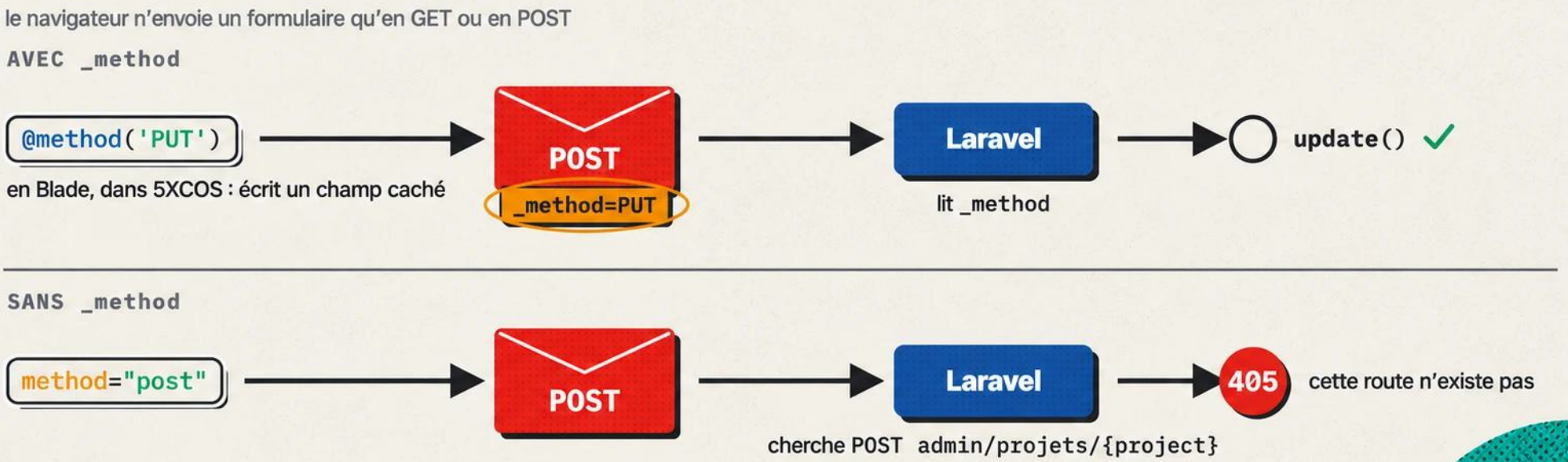
`resources/js/actions/App/Http/Controllers/Admin/ProjectController.ts`
update et destroy y sont depuis le chapitre 1.10 : Wayfinder suit les routes déclarées par Route::resource()

Chaque ligne relie ce que vous écrivez au fichier à ouvrir d'abord. Lisez le modèle, puis adaptez-le à vos projets.

1.13

? Un formulaire HTML n'envoie qu'en GET ou en POST. Comment Laravel reconnaît-il une modification ?

Le formulaire part en POST avec un champ _method : Laravel le lit pour chercher la route PUT ou DELETE.

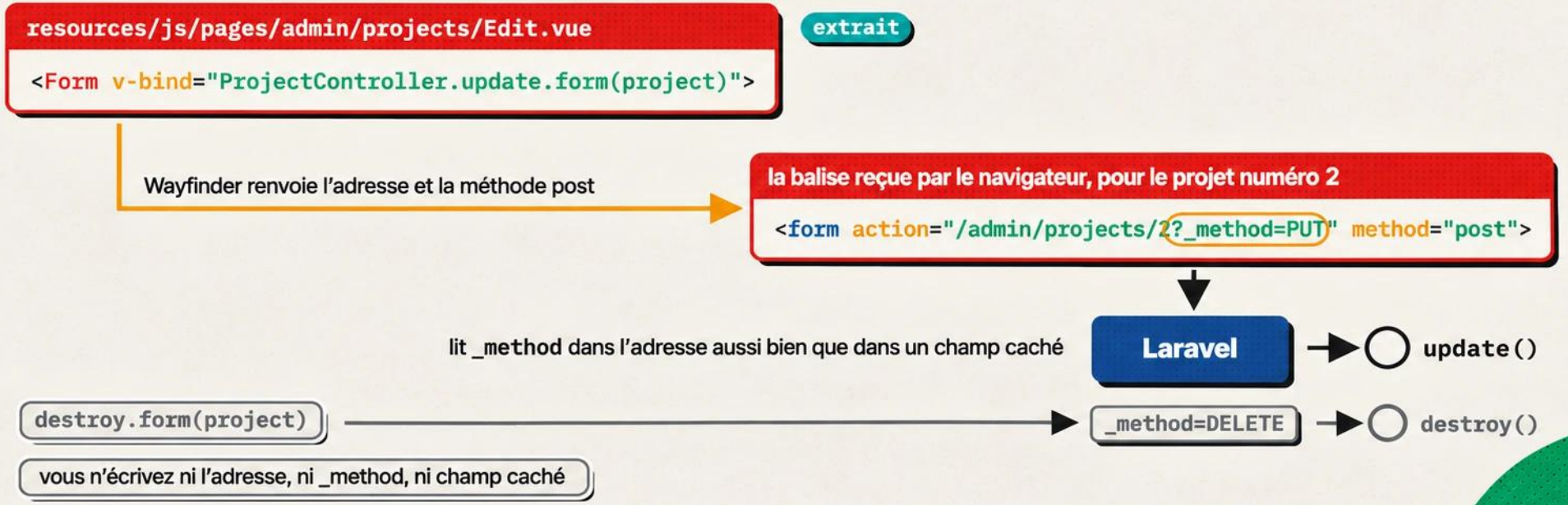


En haut, le champ caché annonce la vraie méthode. En bas, sans lui, Laravel cherche une route POST qui n'existe pas.

1.13

? Avec Wayfinder, vous n'écrivez aucun champ caché. Où se trouve alors _method ?

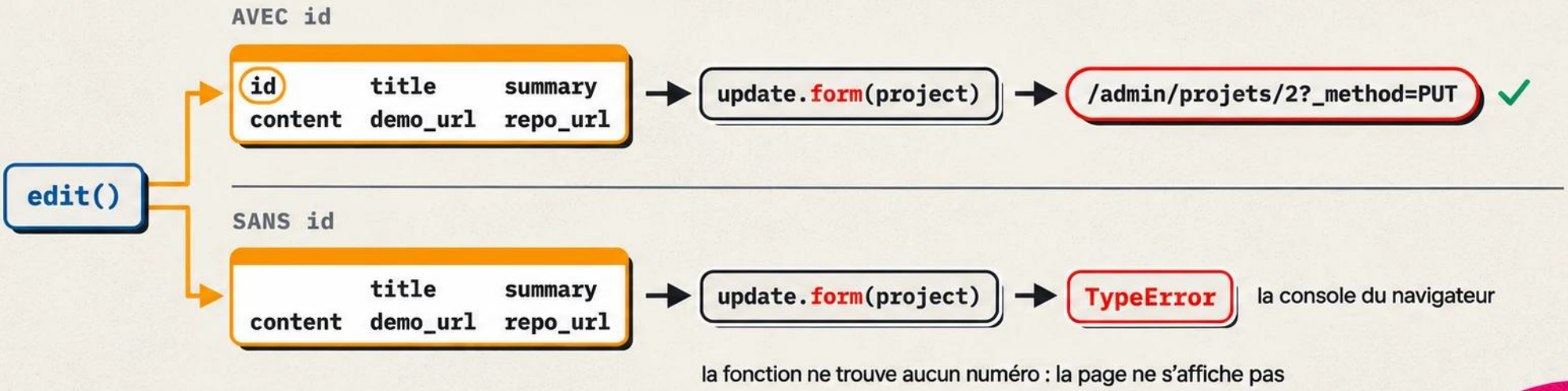
update.form(project) met _method=PUT dans l'adresse du formulaire : Laravel l'y lit comme dans un champ caché.



Suivez la ligne de code jusqu'à la balise : la méthode voyage dans l'adresse, après le point d'interrogation.

? update.form(project) construit l'adresse /admin/projets/2. Où Wayfinder trouve-t-il le 2 ?

Wayfinder lit le numéro dans la clé id du projet : sans id dans les props de edit(), la page ne s'affiche pas.



la signature générée attend un nombre ou un objet `{ id: number }`

Même page, mêmes props, une clé en moins. En bas, Wayfinder ne trouve aucun numéro, et la page ne s'affiche pas.

1.13

? Edit.vue reprend le formulaire de Create.vue. Comment ses cases arrivent-elles déjà cochées ?

:default-value donne à chaque champ sa valeur de départ : une case est cochée si le projet porte cette étiquette.

```
$project->tags()->pluck('tags.id')
```

côté serveur : la liste des numéros d'étiquettes du projet

resources/js/pages/admin/projects/Edit.vue — extrait

```
<Checkbox
    name="tags[]"
    :value="String(tag.id)"
    :default-value="project.tags.includes(tag.id)"
/>
```

la page de modification du Quiz des capitales



un champ non contrôlé reçoit sa valeur de départ par `:default-value`, comme dans la page de profil du kit

La case reçoit `true` ou `false` : cochée si le numéro de l'étiquette figure dans les étiquettes du projet.

1.13

? Le fil d'Ariane affiche le titre du projet. Pourquoi un objet fixe ne suffit-il plus ?

defineOptions ne voit pas les props de la page : donnez-lui une fonction, qu'Inertia appelle avec ces props.

`Create.vue` : un objet fixe, écrit dans `defineOptions` — ne voit pas les props

Inertia appelle la fonction avec les props de la page

`resources/js/pages/admin/projects/Edit.vue` — extrait

```
defineOptions({
    layout: (pageProps) => ({
        breadcrumbs: [
            { title: 'Projets', href: index() },
            { title: pageProps.project.title, href: edit(pageProps.project) },
        ],
    }),
});
```

le fil d'Ariane de la page de modification

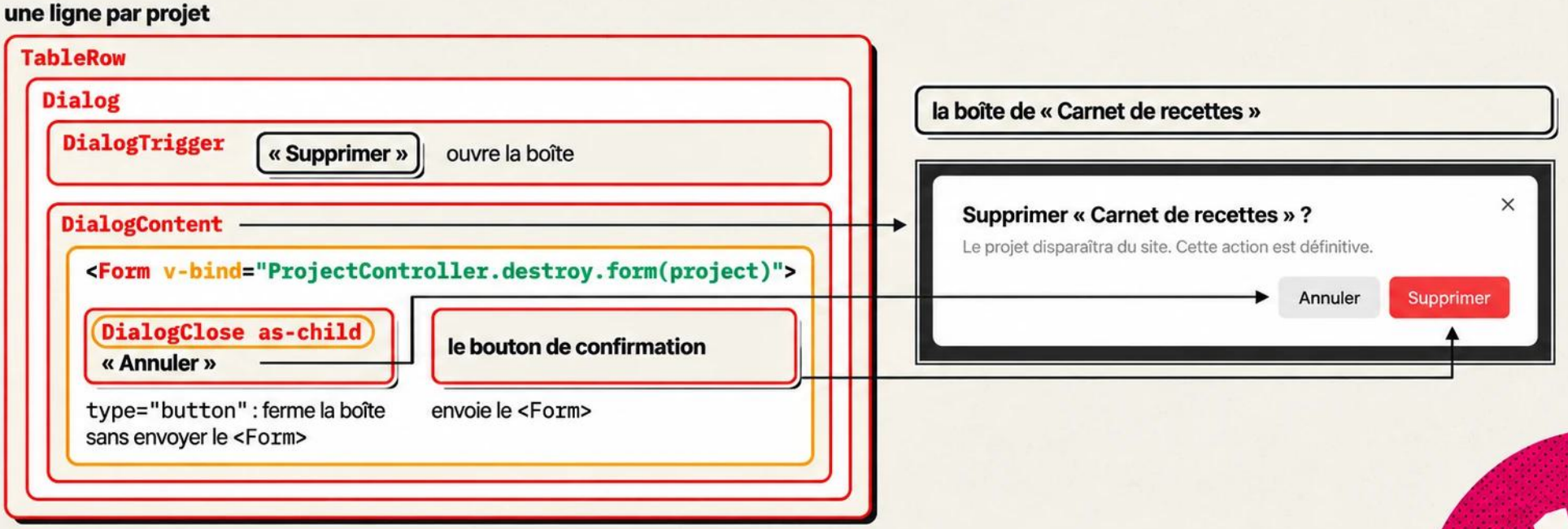
Projets › Quiz des capitales

Modifier le projet

La fonction reçoit les props et renvoie les options du layout. Le titre du projet devient la deuxième étape du fil.

? « Supprimer » ne doit rien effacer au premier clic. Où placer la boîte de confirmation ?

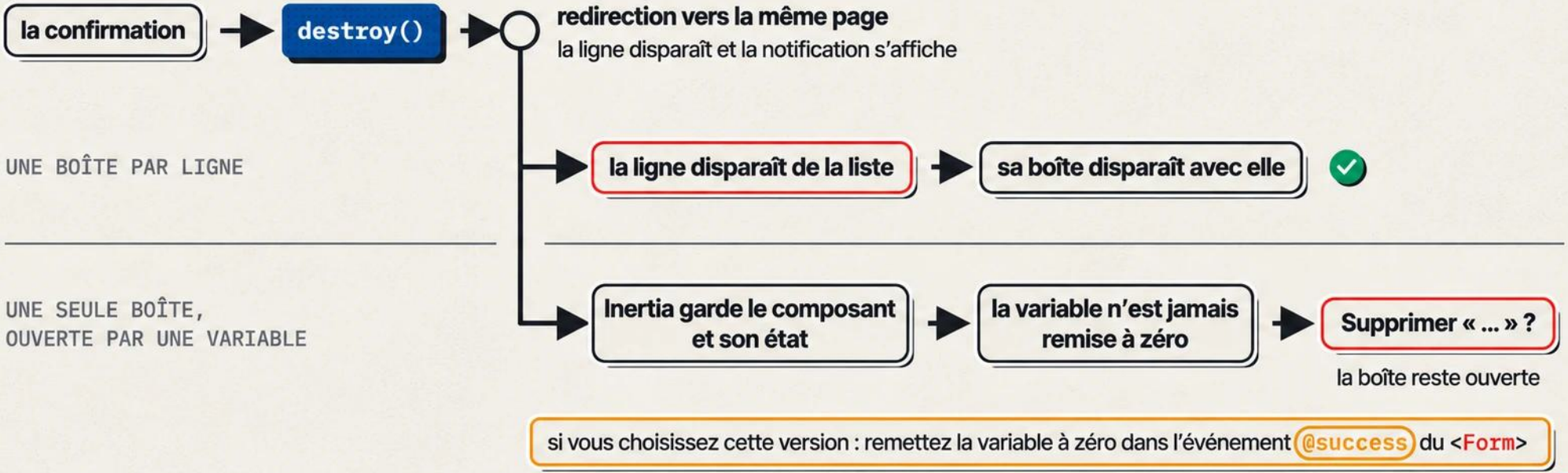
Placez un Dialog dans chaque ligne du tableau, comme DeleteUser.vue : la boîte contient le <Form> de suppression.



Une boîte par projet, posée dans sa ligne. « Annuler » ferme sans envoyer ; seule la confirmation envoie le formulaire.

? Avec une seule boîte placée hors du tableau, pourquoi reste-t-elle ouverte après la suppression ?

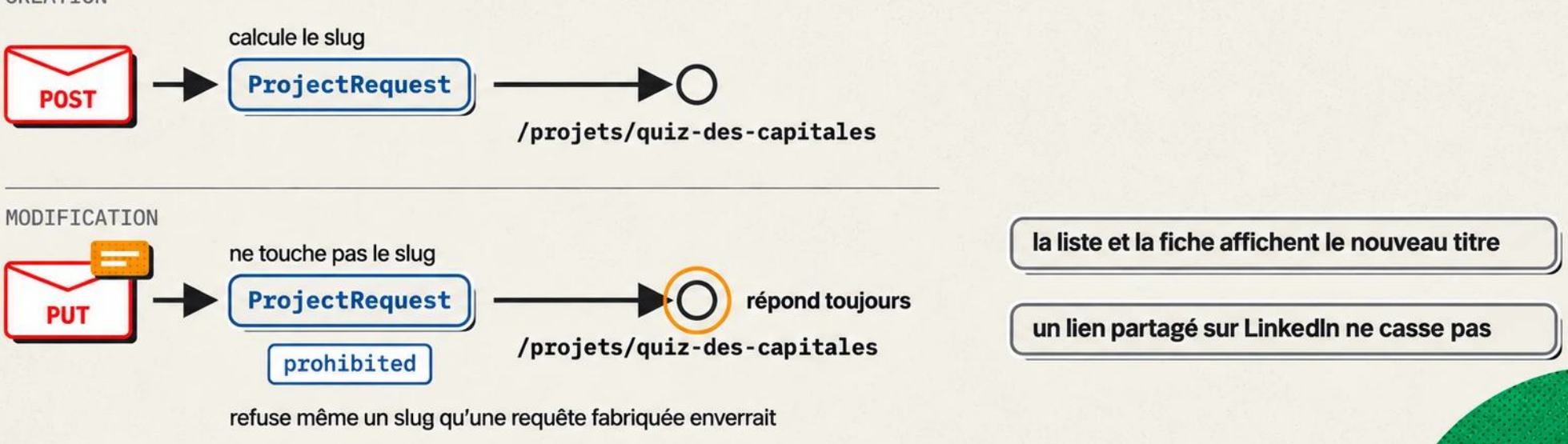
Inertia garde la page après la redirection : la boîte d'une ligne disparaît avec elle, une boîte hors du tableau reste ouverte.



Suivez les deux bandes après la même redirection. En haut, rien à fermer ; en bas, la variable garde la boîte ouverte.

? Vous corrigez le titre du quiz. L'adresse /projets/quiz-des-capitales change-t-elle ?

Non : ProjectRequest ne calcule le slug qu'à la création, et l'adresse publique d'un projet ne change plus.



Les deux bandes arrivent à la même station : le titre change, l'adresse publique reste celle de la création.

? L'envoi échoue. Où regardez-vous en premier, selon ce que vous voyez à l'écran ?

Chaque symptôme a son endroit où lire la cause : l'onglet Réseau, la page d'erreur de Laravel, l'adresse ou la console.

CE QUE VOUS VOYEZ	OÙ REGARDER	CE QUE VOUS Y LISEZ
303 rien ne s'affiche	l'onglet Réseau : la dernière requête vers la page du formulaire	**errors** le champ refusé et son message : ajoutez un InputError
500 une page d'erreur au-dessus de la vôtre	le message, le fichier, la ligne `storage/logs/laravel.log` sous Windows	souvent une faute de frappe, ou une colonne absente de `#[Fillable]`
404 ou un projet vide	l'adresse dans la barre du navigateur	~~`Project $projet`~~ → `Project $project`
une page blanche	la console, avec F12	`TypeError` sur `update.form()` ou `edit()` : un projet sans `id`

avant de demander de l'aide : le message exact, le fichier, et la requête dans l'onglet Réseau

Partez de ce que vous voyez, colonne de gauche, et lisez la rangée jusqu'au bout : l'endroit, puis la cause.

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

À vous, soixante minutes : modifiez d'abord, supprimez ensuite, et vérifiez chaque critère avant de passer à la suite.

À FAIRE

1. **Modifier, environ 35 minutes :** `edit()` et `update()`, puis `Edit.vue`, puis le bouton « Modifier » dans la liste.
2. **Supprimer, environ 25 minutes :** `destroy()`, puis le bouton « Supprimer » et sa boîte de dialogue dans la liste.

Modifiez un vrai projet, puis créez un projet d'essai et supprimez-le.

À VÉRIFIER

60 min

- [ ] Le fil d'Ariane affiche « Projets », puis le titre du projet.
- [ ] Un titre vidé affiche « Le champ titre est obligatoire. », et les autres champs gardent leur valeur.
- [ ] Un titre modifié apparaît dans la liste et sur la fiche, mais l'adresse `/projets/...` ne change pas.
- [ ] Une étiquette décochée disparaît du projet.
- [ ] « Annuler » ferme la boîte sans rien supprimer. La confirmation retire la ligne de la liste.

À LA MAISON

Terminez et cochez chaque critère de la section 1. Terminez votre planche de direction artistique du chapitre 1.09 : elle se remet en PDF avec le portfolio.

Deux parties à gauche, dans l'ordre ; cinq vérifications à droite, dans le navigateur. La maison termine le travail.