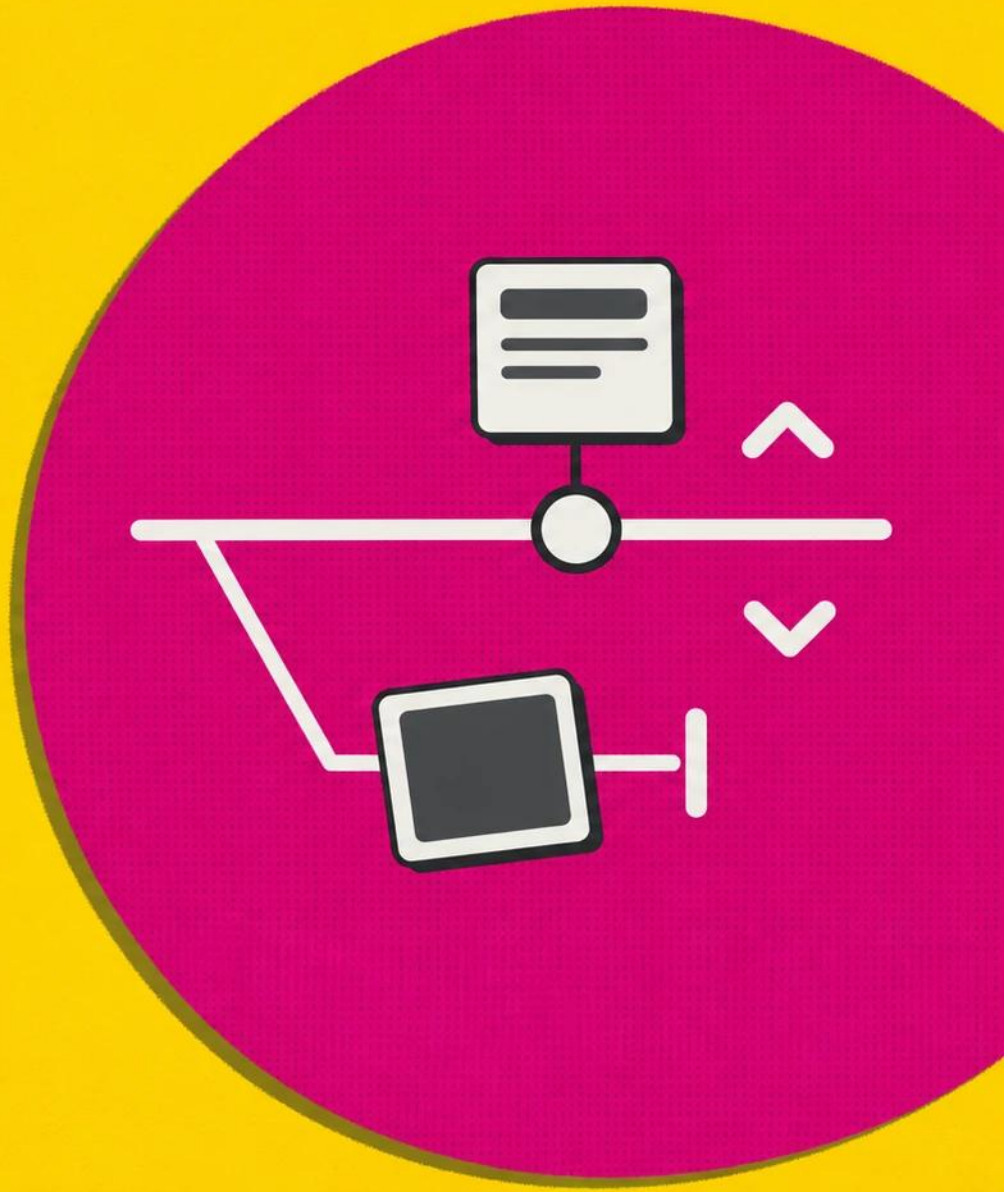
1.16

TEMPS 1 · LE PORTFOLIO · SÉANCE 4

Brouillon ou publié

Thibault Six · IFOSUP Wavre · 2026-2027



? Un projet à moitié écrit doit rester en base sans paraître sur le site. Comment le marquer ?

La colonne published_at sert d'interrupteur : une date veut dire publié, null veut dire brouillon.

la migration du chapitre 1.06

```
$table->timestamp('published_at')->nullable();
```

une date, ou null

database/seeders/ProjectSeeder.php — extrait

```
$recipes = Project::create([
    'title' => 'Carnet de recettes',
    // slug, résumé, contenu, image et liens : inchangés
    'published_at' => null,
    'position' => 3,
]);
```

la table projects

title	published_at	
Météo Wavre	une date	✓ publié
Quiz des capitales	une date	✓ publié
Carnet de recettes	null	brouillon

```
$ php artisan migrate:fresh --seed
```

le site public montre encore les trois projets : rien ne lit la colonne pour l'instant

Une seule colonne décide : une date, le projet est publié ; null, il reste un brouillon, toujours dans la base.

1.16

? La condition « *publié* » servira dans plusieurs pages. Comment éviter de la recopier partout ?

Le scope published() range la condition dans le modèle : la partie publique l'appelle, le back-office lit tous les projets.

app/Models/Project.php — extrait

```php
#[Scope]
protected function published(Builder $query): void
{
    $query->whereNotNull('published_at');
}
```

la condition « publié »

un scope : une condition de requête nommée, rangée dans le modèle

la liste publique commence par le scope



le back-office doit tout montrer : sa méthode index() ne change pas




```
$ php artisan tinker --execute="echo App\Models\Project::published()->pluck('title')->implode(', ');"
Météo Wavre, Quiz des capitales
```

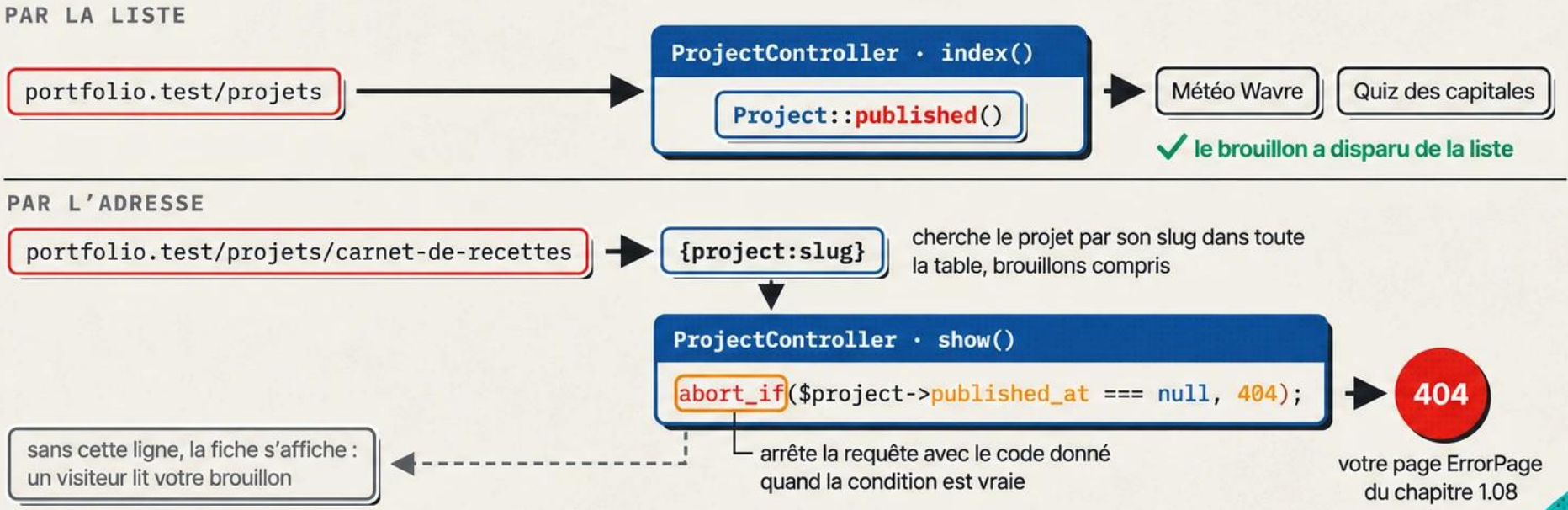
le « Carnet de recettes » n'y est plus

Le scope s'appelle comme une méthode statique du modèle. Tinker le prouve : le brouillon n'y est plus.

1.16

? Le brouillon a disparu de la liste. Un visiteur qui connaît son adresse peut-il encore le lire ?

Oui : la liaison de route ignore le scope, donc show() refuse le brouillon avec abort_if et répond 404.



La liste passe par le scope, l'adresse directe non. Chaque entrée publique vérifie donc l'état elle-même.

? Le back-office montre tous les projets. Comment voir d'un coup d'œil lesquels sont en ligne ?

Une colonne « État » affiche un Badge par ligne : Publié quand published_at a une date, Brouillon sinon.

```
import { Badge } from '@/components/ui/badge';
```

le composant Badge de shadcn-vue : une petite étiquette arrondie

la colonne « État » dans /admin/projets

resources/js/pages/admin/projects/Index.vue extrait

v-if et v-else choisissent l'une ou l'autre selon la date

```
<TableCell>
    <Badge v-if="project.published_at">Publié</Badge>
    <Badge v-else variant="secondary">Brouillon</Badge>
</TableCell>
```

la variante par défaut est foncée

Titre	État
Météo Wavre	Publié
Quiz des capitales	Publié
Carnet de recettes	Brouillon

la variante secondary est grise

index() envoie déjà published_at à la page depuis le chapitre 1.11

Une ligne de code par état : la date choisit le badge. Comparez chaque badge avec la ligne qui le produit.

? Le bouton « Publier » ne change qu'une colonne. Quelle route et quelle méthode lui faut-il ?

Une route PATCH modifie une colonne ; togglePublished() inverse l'interrupteur : une date devient null, null une date.

routes/web.php

```php
Route::patch('projets/{project}/publication', [Admin\ProjectController::class, 'togglePublished'])
    ->name('projects.toggle-published');
```

PATCH : la méthode HTTP d'une modification partielle, une colonne, pas tout le projet

dans le groupe admin : il ajoute admin à l'adresse et admin. au nom

Admin\ProjectController · togglePublished() extrait

```php
$project->update([
    'published_at' => $project->published_at ? null : now(),
]);
```

condition ? valeur si vrai : valeur si faux

LE TOAST RELIT LA COLONNE APRÈS LA MISE À JOUR

publié	null	'Projet repassé en brouillon.'
brouillon	now()	'Projet publié.'

La route donne l'adresse, la méthode bascule la colonne. Lisez chaque rangée : l'état de départ, la nouvelle valeur, le message.

? Le bouton « Publier » n'envoie aucune donnée. Pourquoi le placer quand même dans un <Form> ?

Un <Form> sans champ ne sert qu'à envoyer une requête à la bonne adresse quand on clique sur son bouton submit.

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

```
<Form
    v-bind="ProjectController.togglePublished.form(project)"
    class="inline"
>
    <Button type="submit" variant="outline" size="sm">
        {{ project.published_at ? 'Dépublier' : 'Publier' }}
    </Button>
</Form>
```

aucun champ

Wayfinder génère togglePublished dès que la route et la méthode sont enregistrées : elle porte le nom de la méthode du contrôleur

un clic envoie la requête

le texte du bouton suit l'état du projet

dans la cellule des actions, avant « Modifier »

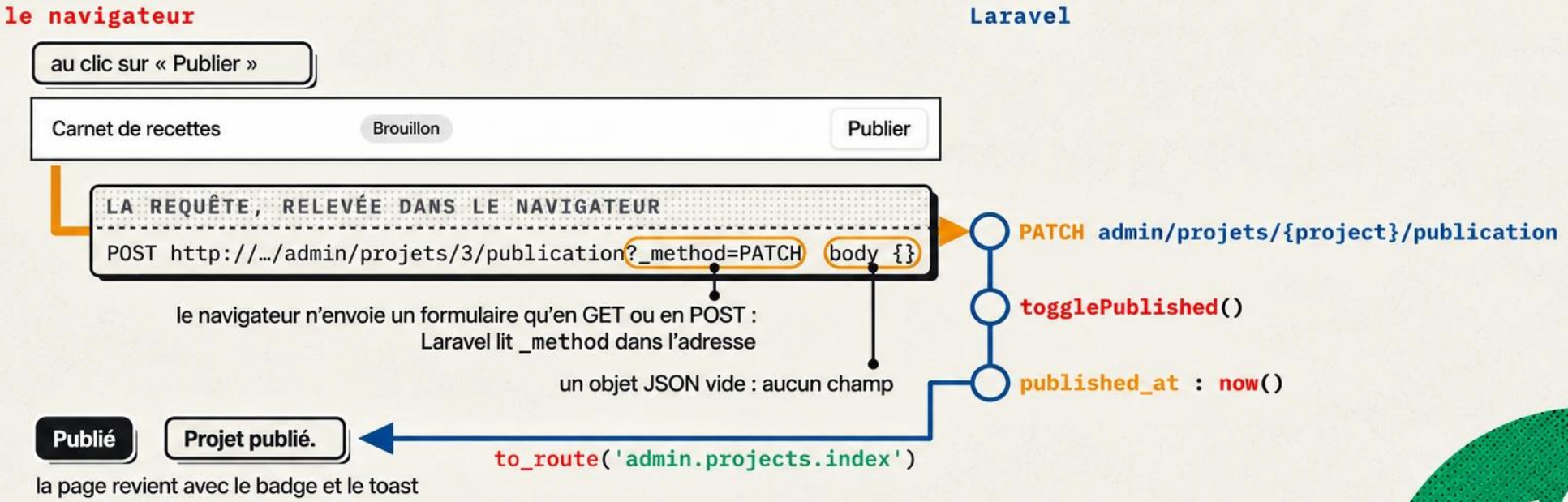
Dépublier	Modifier
Dépublier	Modifier
Publier	Modifier

comme `destroy.form(project)` au chapitre 1.13 : l'objet project doit contenir id, et c'est le cas

Le formulaire ne porte que l'adresse et le bouton. Les deux projets publiés affichent « Dépublier », le brouillon « Publier ».

? Vous cliquez sur « Publier ». Quelle requête part, et quelle méthode du contrôleur répond ?

Wayfinder envoie un POST avec ?_method=PATCH ; Laravel appelle la route PATCH, donc togglePublished().



Suivez le clic : un POST part, Laravel lit _method, la route PATCH appelle togglePublished(), la page revient.

? Pour faire monter « Quiz des capitales » d'un rang, que faut-il changer dans la base ?

Pour changer l'ordre, moveUp() trouve le voisin du dessus et échange sa position avec celle du projet.

AVANT

position	title
1	Météo Wavre
2	Quiz des capitales
3	Carnet de recettes

Monter →

APRÈS

position	title
1	Quiz des capitales
2	Météo Wavre
3	Carnet de recettes

Admin\ProjectController · moveUp() extrait

```php
$previous = Project::where('position', '<', $project->position)
    ->orderByDesc('position')
    ->first();
```

le premier projet n'a pas de voisin du dessus : rien ne se passe

`moveDown()` : la même méthode avec `'>'` et `orderBy('position')`

le voisin du dessus : parmi les projets de position plus petite, celui qui a la plus grande

swapPositions() extrait

```php
[$first->position, $second->position] = [$second->position, $first->position];
```

à gauche du =, les deux propriétés reçoivent, dans l'ordre, les deux valeurs du tableau de droite

Seules deux positions changent, celles des deux voisins. Le tri par position fait le reste, en public et en administration.

? Le premier projet ne peut pas monter. Comment le tableau empêche-t-il ce clic ?

Chaque ligne porte deux <Form> sans champ ; le rang désactive Monter en première ligne, Descendre en dernière.

```
<TableRow v-for="(project, position) in projects" :key="project.id">
```

Vue donne le rang de chaque ligne en second élément : `position` vaut 0 pour la première ligne

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

```
<Button
    type="submit"
    variant="ghost"
    size="icon-sm"
    aria-label="Monter"
    :disabled="position === 0"
```

un bouton qui ne montre qu'une icône porte un `aria-label`, lu par les lecteurs d'écran

Monter, désactivé

```
:disabled="position === projects.length - 1"
```

le même, pour « Descendre »

Descendre, désactivé

Ordre	Titre
↑ ↓	Météo Wavre
↑ ↓	Quiz des capitales
↑ ↓	Carnet de recettes

la colonne « Ordre »

la requête part comme celle de la publication

```
POST http://…/admin/projets/1/descendre?_method=PATCH   body {}
```

Comparez le code et l'écran : Monter sur la première ligne et Descendre sur la dernière sont grisés, désactivés par le rang.

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

À vous, trente minutes : un brouillon caché du site et fermé par une 404, publié d'un clic depuis le back-office.

À FAIRE

1. Un de vos projets en brouillon dans le seeder, puis `php artisan migrate:fresh --seed`.
2. Le scope `published()` dans `Project`, essayé dans Tinker.
3. La liste publique qui n'affiche que les projets publiés.
4. `abort_if` dans `show()`, pour que l'adresse d'un brouillon réponde 404.
5. La colonne « État » avec un `Badge` dans la liste d'administration.
6. La route `PATCH admin.projects.toggle-published`, la méthode `togglePublished()` et son `<Form>`.

À VÉRIFIER

30 min

- [ ] `portfolio.test/projets` ne montre pas votre brouillon.
- [ ] L'adresse de votre brouillon répond par votre page 404.
- [ ] `/admin/projets` montre tous vos projets, avec un badge « Publié » ou « Brouillon ».
- [ ] « Publier » fait apparaître le projet sur le site, avec le toast « Projet publié. », et « Dépublier » le retire.

BONUS

7. Les routes `monter` et `descendre`, `moveUp()`, `moveDown()` et `swapPositions()`.
8. La colonne « Ordre » avec ses deux boutons, désactivés aux extrémités.

- [ ] « Descendre » sur la première ligne l'échange avec la deuxième, dans le back-office et sur le site public.

Six étapes à gauche, quatre vérifications à droite ; le bonus, en bas, attend que les quatre cases soient cochées.

1.16