

Application des surnoms des enseignants avec Adonis - Step9

Pour pouvoir modifier un enseignant, nous allons devoir :

- créer deux routes :
 - une route pour afficher le formulaire permettant de renseigner les informations de l'enseignant
 - une route pour gérer l'ajout de l'enseignant
- ajouter deux méthodes au contrôleur
- ajouter une vue

Routes

Dans le fichier `routes.ts`:

```
// Route permettant d'afficher le formulaire permettant la mise à jour d'un
enseignant
router.get('/teacher/:id/edit', [TeachersController, 'edit']).as('teacher.edit')

// Route permettant la modification de l'enseignant
router.put('/teacher/:id/update', [TeachersController,
'update']).as('teacher.update')
```

Contrôleur

Dans le contrôleur `teachers_controller.ts`:

```
/**
 * Afficher le formulaire permettant la mise à jour d'un enseignant
 */
async edit({ params, view }: HttpContext) {
    // Sélectionner l'enseignant dont on veut mettre à jour des informations
    const teacher = await Teacher.findOrFail(params.id)

    // Récupération des sections triées par le nom
    const sections = await Section.query().orderBy('name', 'asc')

    // Afficher la vue
    return view.render('pages/teachers/edit.edge', {
        title: 'Modifier un enseignant',
        teacher,
        sections,
    })
}

/**
```

```

* Gérer la soumission du formulaire pour la mise à jour d'un enseignant
*/
async update({ params, request, session, response }: HttpContext) {
    // Validation des données saisies par l'utilisateur
    const { gender, firstname, lastname, nickname, origine, sectionId } =
        await request.validateUsing(teacherValidator)

    // Sélectionner l'enseignant dont on veut mettre à jour des informations
    const teacher = await Teacher.findOrFail(params.id)

    // Met à jour les infos de l'enseignant
    teacher.merge({
        gender,
        firstname,
        lastname,
        nickname,
        origine,
        sectionId,
    })

    const teacherUpdated = await teacher.save()

    // Afficher un message à l'utilisateur
    session.flash(
        'success',
        `L'enseignant ${teacherUpdated.lastname} ${teacherUpdated.firstname} a été
mis à jour avec succès !`
    )

    // Redirige l'utilisateur sur la home
    return response.redirect().toRoute('home')
}

```

Ajout de la vue edit.edge

Voici la vue `edit.edge` très semblable à `create.edge` :

```

@component('components/layout', { title: title })
<h3>
    {{ title }}
</h3>

{{-- Notez l'ID dans la route et le spoofing de méthode PUT --}}
<form action="{{ route('teacher.update', { id: teacher.id }) }}?_method=PUT"
method="POST">
    {{ csrfField() }}
    {{-- Bloc Genre (Radio) --}}
    <label>
        <input
            type="radio"
            name="gender"

```

```
        value="man"
        {{ old('gender') || teacher.gender === 'man' ? 'checked' : '' }}
    />
    Homme
</label>
<label>
    <input
        type="radio"
        name="gender"
        value="woman"
        {{ old('gender') || teacher.gender === 'woman' ? 'checked' : '' }}
    />Femme
</label>
<label>
    <input
        type="radio"
        name="gender"
        value="other"
        {{ old('gender') || teacher.gender === 'other' ? 'checked' : '' }}
    />Autre
</label>
@inputError('gender')
<p class="text-danger">
    {{ $messages.join(', ') }}
</p>
@end

{{-- Bloc Prénom --}}
<div class="field">
    <label for="firstname">Prénom</label>
    <input type="text" id="firstname" name="firstname" value="{{
old('firstname') || teacher.firstname }}" />
    @inputError('firstname')
        <p class="text-danger">
            {{ $messages.join(', ') }}
        </p>
    @end
</div>

{{-- Bloc Nom --}}
<div class="field">
    <label for="lastname">Nom</label>
    <input type="text" id="lastname" name="lastname" value="{{ old('lastname')
|| teacher.lastname }}" />
    @inputError('lastname')
        <p class="text-danger">
            {{ $messages.join(', ') }}
        </p>
    @end
</div>

{{-- Bloc Surnom --}}
<div class="field">
    <label for="nickname">Surnom</label>
```

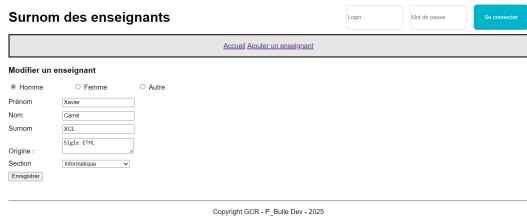
```
<input  
    type="text"  
    id="nickname"  
    name="nickname"  
    value="{{ old('nickname') || teacher.nickname || '' }}"  
/>  
@inputError('nickname')  
    <p class="text-danger">  
        {{ $messages.join(', ') }}  
    </p>  
@end  
</div>  
  
{-- Bloc Origine (Textarea) --}  
<div class="field">  
    <label for="origine">Origine</label>  
    <textarea id="origine" name="origine">{{ old('origine') || teacher.origine }}</textarea>  
    @inputError('origine')  
        <p class="text-danger">  
            {{ $messages.join(', ') }}  
        </p>  
    @end  
</div>  
  
{-- Bloc Section (Select) --}  
<div class="field">  
    <label for="sectionId">Section</label>  
    <select name="sectionId" id="sectionId">  
        <option value="">  
            Choisissez une section  
        </option>  
        @each(section in sections)  
            <option  
                value="{{ section.id }}"  
                {{ (old('sectionId') || teacher.sectionId) == section.id ? 'selected' : '' }}>  
                {{ section.name }}  
            </option>  
        @end  
    </select>  
    @inputError('sectionId')  
        <p class="text-danger">  
            {{ $messages.join(', ') }}  
        </p>  
    @end  
</div>  
  
<button type="submit">Mettre à jour</button>  
</form>  
@end
```

Modifier la vue **home** et la vue **show**

Dans ces 2 deux vues, nous devons mettre à jour le lien pour modifier un enseignant :

```
<a href="{{ route('teacher.edit', {id: teacher.id}) }}>
    
</a>
```

Etat de l'application à la fin de cette étape



The screenshot shows a web form titled "Surnom des enseignants". At the top right are links for "Login", "Mot de passe", and "Se connecter". Below the title is a navigation bar with "Accueil" and "Ajouter un enseignant". The main section is titled "Modifier un enseignant" and contains the following fields:

- Gender selection: Radio buttons for "Homme" (checked), "Femme", and "Autre".
- First Name: Input field containing "Xavier".
- Last Name: Input field containing "Garné".
- Middle Name: Input field containing "XCL".
- Origin: Input field containing "Sigle ETL".
- Section: Select dropdown menu showing "Méthodique".
- Action: A "Enregistrer" button at the bottom.

At the very bottom of the page, there is a small copyright notice: "Copyright GCR - P_Bulle Dev - 2025".

Dans la prochaine étape **step10**, nous allons commencer à mettre en place l'authentification.