

Application des surnoms des enseignants avec Adonis

- Step8

Pour pouvoir ajouter 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 un validateur
- ajouter deux méthodes au contrôleur
- ajouter une vue

Routes

Dans le fichier `routes.ts`:

```
// Route permettant d'afficher le formulaire permettant l'ajout d'un enseignant
router.get('/teacher/add', [TeachersController, 'create']).as('teacher.create')

// Route permettant l'ajout de l'enseignant
router.post('/teacher/add', [TeachersController, 'store']).as('teacher.store')
```

Valdateur

Le validateur va nous permettre de valider les données saisies par l'utilisateur.

Comme d'habitude, nous allons utiliser le CLI pour créer le validateur.

```
$ node ace make:validator TeacherValidator
DONE: create app/validators/teacher.ts
```

Voici maintenant le contenu du fichier `validators/teacher.ts` :

```
import vine from '@vinejs/vine'

const teacherValidator = vine.compile(
  vine.object({
    // Utilisation d'un enum pour le genre
    gender: vine.enum(['woman', 'man', 'other'] as const),

    firstname: vine.string().trim().minLength(2),
    lastname: vine.string().trim().minLength(2),
    nickname: vine.string().trim().minLength(2),
    origine: vine.string().trim().minLength(2),

    // S'assurer que c'est un nombre entier positif
```

```
        sectionId: vine.number().positive(),
    })
}

export { teacherValidator }
```

Contrôleur

Dans le contrôleur `teachers_controller.ts` :

```
/**
 * Afficher le formulaire pour créer un nouvel enseignant
 */
async create({ view }: HttpContext) {
    // Récupération des sections triées par le nom
    const sections = await Section.query().orderBy('name', 'asc')

    // Appel de la vue
    return view.render('pages/teachers/create', { title: "Ajout d'un enseignant",
sections })
}

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

    // Création du nouvel enseignant
    const teacher = await Teacher.create({ gender, firstname, lastname, nickname,
origine, sectionId })

    // Afficher un message à l'utilisateur
    session.flash('success', `Le nouvel enseignant ${teacher.lastname}
${teacher.firstname} a été ajouté avec succès !`)

    // Rediriger vers la homepage
    return response.redirect().toRoute('home')
}
```

Rappel : après avoir copier/coller le code, vous aurez besoin de faire les imports.

Pour cela, vous aurez besoin du raccourci ctrl + espace précédemment expliqué.

Ajout de la vue create.edge

Cette vue a pour but d'afficher le formulaire permettant de renseigner les informations de l'enseignant.

Si certaines règles ne sont pas respectés (champs obligatoires, nombre de caractères, etc) on doit afficher des erreurs à l'utilisateur directement dans le formulaire.

Ces composants seront utilisés dans la vue `create.edge` de cette manière :

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

<form action="{{ route('teacher.store') }}" method="POST">
  {{ csrfField() }}
  {{-- Bloc Genre (Radio) --}}
  <label>
    <input type="radio" name="gender" value="man" {{ old('gender') === 'man' ?
'checked' : '' }} />Homme
  </label>
  <label>
    <input type="radio" name="gender" value="woman" {{ old('gender') === 'woman'
? 'checked' : '' }} />Femme
  </label>
  <label>
    <input type="radio" name="gender" value="other" {{ old('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') || '' }}" />
    @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')
|| '' }}" />
    @inputError('lastname')
    <p class="text-danger">
      {{ $messages.join(', ') }}
    </p>
  </div>
```

```

    @end
  </div>

  {{-- Bloc Surnom --}}
  <div class="field">
    <label for="nickname">Surnom</label>
    <input type="text" id="nickname" name="nickname" value="{{ old('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') || '' }}</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') == section.id ?
'selected' : '' }}>
          {{ section.name }}
        </option>
      @end
    </select>
    @inputError('sectionId')
    <p class="text-danger">
      {{ $messages.join(', ') }}
    </p>
  @end
</div>

  <button type="submit">Enregistrer</button>
</form>
@end

```

Modifier le header

Dans le header, nous devons mettre à jour le lien pour ajouter un nouvel enseignant.

```
<nav>
  <a href="{ route('home') }">Accueil</a>&nbsp;
  <a href="{ route('teacher.create') }">Ajouter un enseignant</a>
</nav>
```

Message de validation en francais

Nos messages de validation sont en anglais !

Surnom des enseignants

 Login Mot de passe

[Accueil](#) [Ajouter un enseignant](#)

Ajout d'un enseignant

☐ Homme ☐ Femme ☐ Autre

The gender field must be defined

Prénom

The firstame field must be defined

Nom

The lastname field must be defined

Surnom

The nickname field must be defined

Origine :

The origine field must be defined

Section

The sectionid field must be defined

Copyright GCR - P_Bulle Dev - 2025

Nous allons faire en sorte qu'ils soient affichés en francais.

Pour commencer on doit installer un nouveau package :

```
node ace add @adonisjs/i18n
```

Ensuite on change la `defaultLocale` dans le fichier `config\i18n.ts`

```
import app from '@adonisjs/core/services/app'
import { defineConfig, formatters, loaders } from '@adonisjs/i18n'

const i18nConfig = defineConfig({
  defaultLocale: 'fr',
  formatter: formatters.icu(),

  loaders: [
    /**
     * The fs loader will read translations from the
     * "resources/lang" directory.
     *
     * Each subdirectory represents a locale. For example:
     * - "resources/lang/en"
     * - "resources/lang/fr"
     */
  ]
})
```

```

    * - "resources/lang/it"
    */
    loaders.fs({
      location: app.languageFilePath(),
    }),
  ],
})

export default i18nConfig

```

Ensuite, on définit un fichier `validator.json` dans le répertoire `ressources\lang\fr`

```

{
  "shared": {
    "fields": {
      "gender": "Genre",
      "firstname": "Prénom",
      "lastname": "Nom",
      "nickname": "Surnom",
      "origine": "Origine",
      "sectionId": "Section"
    },
    "messages": {
      "required": "Le champ {field} est obligatoire.",
      "minLength": "Le champ {field} doit contenir au moins { min } caractères.",
      "email": "Le champ doit {field} être une adresse e-mail valide."
    }
  }
}

```

Après avoir redémarrer le serveur, vous devriez avoir les messages en français :

Surnom des enseignants Login Mot de passe

[Accueil](#) [Ajouter un enseignant](#)

Ajout d'un enseignant

☐ Homme ☐ Femme ☐ Autre

Le champ Genre est obligatoire.

Prénom

Le champ Prénom doit contenir au moins 2 caractères.

Nom

Le champ Nom est obligatoire.

Surnom

Le champ Surnom est obligatoire.

Origine

Le champ Origine est obligatoire.

Section

Le champ Section est obligatoire.

Copyright GCR - P_Bulle Dev - 2025

Prochaine étape

Dans la prochaine étape [step9](#), nous allons gérer la mise à jour d'un enseignant.