

# Application des surnoms des enseignants - Step1

Dans cette étape, nous allons travailler dans une vue (le V du pattern MVC).

## Qu'est qu'une vue ?

Une vue est un fichier qui contient du HTML combiné avec des balises dynamiques pour insérer des données générées par le serveur.

Ces fichiers sont généralement gérés par un moteur de templates, tel que Edge dans AdonisJS.

## Version statique de l'application des surnoms des enseignants

Pour commencer cette aventure avec le framework AdonisJS, nous allons récupérer la [Version statique HTML5 / CSS3](#) de l'application des surnoms des enseignants.

Comme son nom l'indique, cette version statique ne contient que des fichiers HTML, CSS et 1 fichier JS (mais exécuté côté client).

Au cours de la construction de notre application, nous allons récupérer du HTML de cette version statique.

Cela nous permettra de nous concentrer sur l'apprentissage d'AdonisJS et pas sur le HTML5.

## Fichier css

Nous allons copier le CSS ci-dessous pour le coller dans le fichier `app.css` présent dans le répertoire `resources/css`.

```
body {  
    font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  
    margin: auto;  
    width: 80%;  
}  
  
nav {  
    border: black solid 3px;  
    background-color: hsl(0, 0%, 90%);  
    height: 50px;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}  
  
.container {  
    height: 50%;  
}  
  
footer {  
    border-top: black solid 1px;  
    text-align: center;  
}
```

```
margin-top: 3vw;
}

table {
  width: 100%;
  border-collapse: collapse;
  text-align: left;
}

th,
td {
  padding: 1vw;
  border-bottom: #ddd solid 1px;
}

.containerOptions a,
.user-head a {
  text-decoration: none;
  height: 100%;
  margin-left: 1vw;
}

.user-head {
  width: 100%;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
}

.user-footer {
  text-align: right;
}

.left,
.right {
  display: inline-block;
}

.left {
  width: 70%;
}

label {
  display: inline-block;
  width: 10%;
}

.container-header {
  display: flex;
  justify-content: center;
  align-items: center;
  padding-top: 5%;
}
```

```
.titre-header {
    width: 60%;
    height: 100%;
}

.login-container {
    width: 40%;
    height: 100%;
    display: flex;
    justify-content: end;
    align-items: center;
}

.login-container form {
    display: flex;
    justify-content: space-between;
}

.login-container label {
    display: none;
}

.login-container input,
.login-container .btn-login {
    width: 30%;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 8px;
}

.login-container h3 {
    display: inline-block;
    width: 30%;
    text-align: right;
    margin-right: 1vw;
}

.login-container input {
    height: 4em;
    margin-bottom: auto;
    margin-top: auto;
}

.btn-login {
    background-color: #00b4cc;
}

.btn {
    text-align: center;
    color: white;
    width: 100%;
    border: none;
    cursor: pointer;
}
```

```
height: 4.5em;
border-radius: 8px;
}

.btn:hover {
  opacity: 0.8;
}

.btn-logout {
  background-color: coral;
  padding: 20px;
  margin: 10px;
}

.text-danger {
  color: red;
}

.alert {
  padding: 15px;
  border-radius: 4px;
  font-size: 16px;
  margin: 10px 0;
}

.alert-success {
  background-color: #d4edda;
  border: 1px solid #c3e6cb;
  color: #155724;
}

.alert-danger {
  background-color: #f8d7da;
  border: 1px solid #f5c6cb;
  color: #721c24;
  padding: 15px;
  border-radius: 4px;
  font-size: 16px;
  margin: 10px 0;
}

.field {
  margin-top: 10px;
  margin-bottom: 10px;
}

.radio {
  margin-top: 5px;
  margin-bottom: 15px;
}

.containerOptions a,
.containerOptions form,
.user-head a {
```

```
text-decoration: none;  
height: 100%;  
margin-left: 1vw;  
}
```

## Modifier la vue 'home'

Nous allons modifier la vue `home.edge` pour copier/coller le html présent dans le fichier 'index.html' de la version statique.

Ensuite, nous faisons un léger changement dans le header pour la balise `nav` :

```
<nav>  
  <a href="index.html">Accueil</a>&nbsp;  
  <a href="addTeacher.html">Ajouter un enseignant</a>  
</nav>
```

Import des fichiers statiques css et js Dans la balise `head` nous allons remplacer la ligne

```
<link href=".css/style.css" rel="stylesheet" />
```

par

```
@vite(['resources/css/app.css', 'resources/js/app.js'])
```

afin d'importer les fichiers statiques.

La directive `@vite` gère automatiquement l'inclusion des scripts nécessaires, que ce soit en mode développement avec le serveur `Vite` ou en production avec les fichiers compilés.

## Images

Dans la version statique de l'application, il y a quelques icônes présents dans le répertoire `img/`.

Dans notre projet AdonisJS, nous allons créer un répertoire `public/` à la racine du projet et copier le répertoire `img/` à l'intérieur.

## Etat de l'application à la fin de cette étape

Pour voir le résultat des modifications effectuées, nous devons commencer par lancer le serveur de développement.

```
npm run dev
```

```
[ info ] starting HTTP server...
[ info ] Server address: http://localhost:3333
[ info ] Watch Mode: HMR
[ info ] Ready in: 1.63 s
[14:51:08.788] INFO (14664): started HTTP server on localhost:3333
```

Notre application a pour l'instant qu'une page à savoir la homepage.

De plus, pour l'instant tout est statique.

Surnom des enseignants

Login Mot de passe Se connecter

Accueil Ajouter un enseignant

Liste des enseignants

Nom	Surnom	Options
Girardet Alain	AGT	edit delete search
Mveng Antoine	AMG	edit delete search
Charmier Grégory	GregLeBarbar	edit delete search

Copyright GCR - bulle web-db - 2022

## Prochaine étape

Dans la prochaine étape [step2](#), nous allons découvrir plus en détail le gestionnaire de template d'AdonisJS : [EDGE](#).