

Application des surnoms des enseignants avec Adonis - Step12

Nous allons mettre en place le logout.

pour cela, nous avons besoin :

- d'une nouvelle route
- d'une nouvelle méthode dans `AuthController`

Routes

```
// Route permettant de se déconnecter
router.post('/logout', [AuthController, 'logout']).as('auth.logout')
```

Contrôleur AuthController

```
/**
 * Gérer la déconnexion d'un utilisateur
 */
async logout({ auth, session, response }: HttpContext) {
  // Utilise le Guard 'web' pour déconnecter l'utilisateur -> Voir le fichier
  config/auth.ts
  await auth.use('web').logout()

  // Affiche un message à l'utilisateur
  session.flash('success', "L'utilisateur s'est déconnecté avec succès")

  // Redirige la réponse sur la route 'home'
  return response.redirect().toRoute('home')
}
```

Bouton 'Se connecter' et 'Se déconnecter'

Dans le fichier `header.edge` nous allons ajouter un bouton "Se déconnecter" ainsi que l'affichage du username de l'utilisateur connecté.

```
...
@if(auth.isAuthenticated)
  {{ auth.user.username }} ({{ auth.user.isAdmin ? 'admin' : 'user' }})
  <form action="{{ route('auth.logout') }}" method="post">
    {{ csrfField() }}
    <button type="submit" class="btn btn-logout">Se déconnecter</button>
  </form>
```

@else

...

Prochaine étape

Dans la prochaine étape [step13](#), nous allons poursuivre avec la gestion des rôles et la protection des pages.