









# Développement WEB PHP - Séance 9

**BUT Informatique parcours DACS**



# Table of contents:

-  | TD 8 : Catalogue d'artistes et de musiques
  - 1.  Mise en route (20 min)
    - 1.1. Création du projet Laravel
    - 1.2. Configuration de la base de données
  - 2.  Modélisation du domaine (25 min)
    - 2.1. Création des modèles et migrations
    - 2.2. Configuration des migrations
  - 3.  Définition des relations Eloquent (10 min)
  - 4.  Création des contrôleurs et des routes (30 min)
    - 4.1. Création des contrôleurs
    - 4.2. Configuration des routes
  - 5.  Vue : Liste des artistes (30 min)
    - 5.1. Contrôleur
    - 5.2. Vue Blade
  - 6.  Création d'un artiste (40 min)
    - 6.1. Enregistrement
  - 7.  Affichage détaillé d'un artiste (40 min)
    - 7.1. Contrôleur
    - 7.2. Vue Blade
  - 9. Structure HTML commune (10 min)
  - 10. Extensions possibles (optionnel)



# | TD 8 : Catalogue d'artistes et de musiques

## 1. 🚦 Mise en route (20 min)

### 1.1. Création du projet Laravel

<https://laravel.com/docs/12.x/installation>

1. Installer Laravel si ce n'est pas déjà fait.
2. Créer un nouveau projet nommé `music-app`.
3. Lancer le serveur de développement Laravel.
4. Vérifier que le site s'ouvre dans le navigateur <http://localhost:8000>

### 1.2. Configuration de la base de données

<https://laravel.com/docs/12.x/database>

5. Configurer le fichier d'environnement `.env` pour pointer vers votre base de données.  
Nous utiliserons sqlite ([lien](#)).
6. Tester la connexion en lançant la commande de migration par défaut fournie avec Laravel.

```
php artisan migrate
```

## 2. 💡 Modélisation du domaine (25 min)

<https://laravel.com/docs/12.x/eloquent>

Nous allons gérer deux entités : **Artist** et **Song**.

### 2.1. Création des modèles et migrations

7. Générer un modèle **Artist** avec migration.
8. Générer un modèle **Song** avec migration.

### 2.2. Configuration des migrations

Modifier la migration d'artistes : 9. Ajouter un champ `name` (string). 10. Ajouter un champ `country` (string, nullable).

Modifier la migration des musiques : 11. Ajouter un champ `title` (string). 12. Ajouter un champ `year` (integer, nullable). 13. Ajouter un champ `duration` (integer, nullable). 14. Ajouter une clé étrangère vers l'artiste associé. `artist_id` (foreignId, onDelete Cascade)

15. Lancer toutes les migrations.

```
php artisan migrate
```

### 3. Définition des relations Eloquent (10 min)

<https://laravel.com/docs/12.x/eloquent-relationships>

Dans le modèle **Artist** : 16. Définir la relation "un artiste possède plusieurs chansons".

Dans le modèle **Song** : 17. Définir la relation "une chanson appartient à un artiste".

---

## 4. Création des contrôleurs et des routes (30 min)

### 4.1. Création des contrôleurs

<https://laravel.com/docs/12.x/controllers>

18. Générer un contrôleur *ArtistController*.

19. Générer un contrôleur *SongController*.

### 4.2. Configuration des routes

<https://laravel.com/docs/12.x/routing>

20. Créer une route d'accueil redirigeant vers la liste des artistes.

21. Définir les routes GET de base pour les artistes `/artists`.

22. Définir les routes GET de base pour les musiques `/songs`.

---

## 5. Vue : Liste des artistes (30 min)

### 5.1. Contrôleur

23. Dans *ArtistController*, implémenter une méthode permettant de récupérer tous les artistes et de les envoyer à une vue.

### 5.2. Vue Blade

<https://laravel.com/docs/12.x/blade>

Créer une vue affichant la liste des artistes : 24. Afficher tous les artistes sous forme de liste. 25. Ajouter un lien vers la page d'ajout d'un artiste. 26. Faire en sorte que chaque nom d'artiste renvoie vers sa page de détails.

---

## 6. 🧑‍🎨 Création d'un artiste (40 min)

### 6.1. Enregistrement

29. Ajouter une opération permettant de créer un nouvel artiste en base de données (route + méthode dans le contrôleur).
  30. Ajouter une méthode dans le contrôleur pour valider les données reçues ([lien](#)).
  31. Rediriger l'utilisateur vers la liste des artistes après la création.
-



## 7. Affichage détaillé d'un artiste (40 min)

### 7.1. Contrôleur

32. Créer une méthode pour afficher la page détaillée d'un artiste.

33. Charger ses musiques associées.

### 7.2. Vue Blade

Créer une vue pour afficher un artiste : 34. Afficher le nom et le pays. 35. Lister les musiques associées (titre + année). 36. Afficher un message si l'artiste n'a pas encore de musique. 37. Ajouter un lien "Ajouter une musique" préremplissant l'artiste.

---

## 9. Structure HTML commune (10 min)

- 44. Créer une vue *layout* contenant l'en-tête et la navigation du site.
  - 45. Adapter les autres vues pour qu'elles héritent de ce layout.
-

## 10. Extensions possibles (optionnel)

- 46. Ajouter la modification d'un artiste.
- 47. Ajouter la suppression d'un artiste.
- 48. Ajouter la modification d'une musique.
- 49. Ajouter la suppression d'une musique.
- 50. Mettre en place une pagination sur la liste des artistes.
- 51. Ajouter un système de recherche.
- 52. Ajouter une API publique permettant de récupérer les artistes en JSON.
- 53. Ajouter des Seeders pour remplir automatiquement la base.