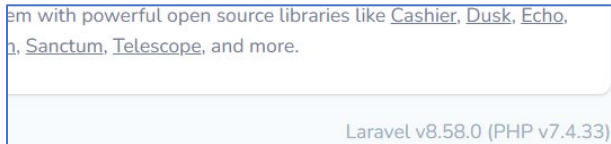


TP1 – Laravel « Maisonneuve » / septembre 2023

Alain Thérout — 2195547

En local pour développer, je roule la même chose que sur Webdev
(sauf légère différence mon php 7.4.33, sur webdev c'est 7.53)



em with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#),
[Sanctum](#), [Telescope](#), and more.

Laravel v8.58.0 (PHP v7.4.33)

Votre base de données initiale aura 2 tables :

Étudiant (id, nom, adresse, phone, email, date de naissance, ville_id)
et **ville** (id, nom).

1. En utilisant les lignes de commande, créer un nouveau projet Laravel
nommée Maisonneuve{votre matricule} (1 pt)

composer create-project --prefer-dist laravel/laravel **Maisonneuve2195547 "8.*"**

2. En utilisant les lignes de commande, créer les modèles (1 pts)

php artisan make:model Etudiant -m

php artisan make:model Ville -m

3. En utilisant les lignes de commande, créer les tables (2 pts)

(elles se sont créées en mettant mon -m à la commande précédente). Mais la recette serait :

php artisan make:migration create_Etudiants_table

php artisan make:migration create_Villes_table

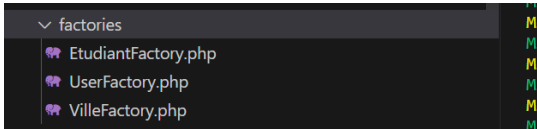
➔ J'ai aussi réglé le problème de **l'erreur 1071** en allant dans Providers faire les modifications vues en classe. J'ai ensuite du jeter la bdd (remplie partiellement) et la recréer. Mon php artisan migrate a généré le bon contenu de bdd

4. En utilisant les lignes de commande, saisir 100 nouveaux étudiants (1 pts)

__Création d'abord des factories

```
php artisan make:factory EtudiantFactory -m Etudiant
```

```
php artisan make:factory VilleFactory -m Ville
```



__ Création des champs dans les factories, puis intégration dans les bases :

```
php artisan tinker
```

J'obtiens :

```
Psy Shell v0.10.8 (PHP 7.4.33 — cli) by Justin Hileman
```

```
>>>
```

puis je tape :

```
// \App\Models\Etudiant::factory()->times(100)->create();
```

Mes étudiants ont été ainsi créés et intégrés dans la bdd:

```
},
App\Models\Etudiant {#3505
  nom: "Cronin",
  prenom: "Adrianna",
  adresse: "666 Adaline Garden",
  telephone: "1-814-695-8341",
  courriel: "mkerluke@example.org",
  naissance: "1976-02-29",
  updated_at: "2023-09-20 13:41:57",
  created_at: "2023-09-20 13:41:57",
  id: 11,
}
```

(...)

94	2023-09-20 13:41:57	2023-09-20 13:41:57	Harris	Ozella	807 Kilback Oval Suite 955	423.608.1681	nbartell@example.org
95	2023-09-20 13:41:57	2023-09-20 13:41:57	O'Hara	Roel	962 Baylee Throughway Suite 119	+1-541-295-2373	ratke.gino@example.com
96	2023-09-20 13:41:57	2023-09-20 13:41:57	Gleason	Lavina	851 Stefanie Stream	+19475608119	edgardo.koepp@example.net
97	2023-09-20 13:41:57	2023-09-20 13:41:57	Boyle	Paula	4991 Senger Viaduct	820-990-0656	jeffrey.prohaska@example.com
98	2023-09-20 13:41:57	2023-09-20 13:41:57	Lubowitz	Jace	25323 Samantha Wall Apt. 106	+1 (757) 693-7613	prohaska.sean@example.com
99	2023-09-20 13:41:57	2023-09-20 13:41:57	Schaefer	Arnaldo	29311 Jonatan Station	620-407-1044	ardella65@example.org
100	2023-09-20 13:41:57	2023-09-20 13:41:57	Carroll	Janice	18339 Schumm Ferry Apt. 181	831-743-2609	geoffrey.dooley@example.com

5. En utilisant les lignes de commande, saisir 15 nouvelles villes (1 pts)

```
// \App\Models\Ville::factory()->times(15)->create();
```

Pour les questions 4 et 5, effectuez une recherche des propriétés de "Factory" pour remplir des valeurs telles que des noms, des adresses, des téléphones, etc. (pas de phrases ou de texte aléatoires).

✓ Fait

6. En utilisant les lignes de commande, créer les contrôleurs (1 pts)

php artisan make:controller Etudia make:controller EtudiantController -m Etudiant

php artisan make:controller VilleController -m Ville

```
PS C:\Users\bobbion\Desktop\00_Sessi
make:controller EtudiantController
Controller created successfully.
make:controller VilleController -m
Controller created successfully.
PS C:\Users\bobbion\Desktop\00_Sessi
```

7. Créez votre layout.blade avec vous CSS, vous devez importer bootstrap (ou du CSS personnalisée) et le concevoir selon vos préférences. (1 pts)

npm install bootstrap@5.3.1

Nommage des fichiers avec dossiers destinés au CRUD tels que vus en classe

8. Travailler avec bootstrap (ou du CSS personnalisée) pour respecter les concepts d'ergonomie, soyez créatif (1pts).

✓ Fait

9. Créer un contrôleur "index" et une vue, pour afficher tous les étudiants, avec un lien pour sélectionner l'étudiant et le mettre à jour. (2 pts)

✓ Fait

10. Créer un contrôleur "create" et une vue, pour saisir un nouvel étudiant. Le formulaire doit avoir un champ "select" avec toutes les villes qui viennent de la base de données. (2 pts)

Un intégration-test a réussi pour 101^e et un 102^e étudiant :

	id	created_at	updated_at	nom	prenom	adresse	telephone	courriel	naissance
<input type="checkbox"/> Éditer Copier Supprimer	101	2023-09-20 17:02:28	2023-09-20 17:02:28	asdf	wwee	555fff	2345	alain.theroux@gmail.com	1999-11-11

wwee asdf	555fff	alain.theroux@gmail.com
Alain Thérour	555fff	ccc@gmail.com

(suite 10)

Malheureusement pour le select des villes, bien que je l'ai créé pour montrer que je sais faire, ma bdd a été mal conçue pour l'intégration d'une ville — donc non-fonctionnel dans le site.

```
<div class="form-group mb-3">
  <!-- <select class="form-select" name="ville_id">
    <option selected>Déterminer la ville</option>
    @foreach($villes as $ville)
      <option value="{{ $ville->id }}">{{ $ville->nom }}</option>
    @endforeach
  </select> -->
</div>
```

11. Créer un contrôleur “show” et une vue, pour afficher un étudiant sélectionné.(2 pts)

✓ Fait

12. Créer un contrôleur “edit” et une vue, pour afficher un étudiant sélectionné dans un formulaire et le mettre à jour. (2 pts)

✓ Fait (avec PUT)

created_at	updated_at	nom	pre nom	adresse	telephone	courriel	ni
2023-09-20 13:41:57	2023-09-20 17:39:35	Conroy	Vidal Sassoun	635 Mills Run	(850) 407-6868	rodriguez.blaney@example.net	15

Avec mention confirmant l'action sous l'information modifiée :

Détails

Nom : Omerio Jacobi

Adresse : 27184 Brandi Knolls

Téléphone : +1 (831) 432-9390

Courriel : qheathcote@example.org

Date de naissance : 1988-12-27

Modifier

Revenir à la une

Supprimer de la base

Données mises à jour

13. Créer un contrôleur “destroy” pour supprimer un étudiant sélectionné. (1 pt)

✓ Fait

Avec mention sur la une grâce à {{session('success')}}

Donnée effacée

Bottin c

14. Publier votre projet dans GitHub (publique) et envoyer le lien dans la documentation. (1 pts)

<https://github.com/therouxalain/etudiants>

15. Enregistrez le projet avec une extension ZIP et ajouter la documentation dans la racine (1pt)

✓ Fait

■ Mise en ligne sur WebDev

✓ Fait <https://e2195547.webdev.cmaisonneuve.qc.ca/Maisonneuve2195547/>