



Examen régional de fin de module  
Année de Formation 2022/2023

Nom : .....

Prénom : .....

Groupe : .....

Etablissement : .....

Réservé à l'établissement Code : .....

Code module: M205

Intitulé du module : Développement back-end

Filière	:	Développement Digital option Web Full Stack	Durée	: 2h
Année	:	2A	Note finale	: / 40
Nom & Prénom du correcteur			Émargement	

**Théorie : (10 Points)**

Sélectionnez la bonne réponse parmi les options proposées pour chaque question :

1- Quelle est la commande pour afficher la liste de toutes les routes ?

(01 Pt)

- ☐ php artisan list
- ☐ php artisan route:list
- ☐ php artisan route-show
- ☐ php artisan route:view

2- Comment créer un middleware avec le nom "middleOne" ?

(01 Pt)

- ☐ php artisan make:middleware middleOne
- ☐ php artisan make:middle middleOne
- ☐ php artisan generate:middleware middleOne
- ☐ php artisan create:middleware middleOne

3- Comment créer un modèle avec le nom "User" ?

(01 Pt)

- ☐ php artisan create:model User
- ☐ php artisan make:model User
- ☐ php artisan new:model User
- ☐ php artisan generate:model User



(02 Pts)

4- Comment créer un contrôleur avec une action unique ("TestController") ?

- ☐ php artisan create:controller TestController -i
- ☐ php artisan make:controller TestController --single
- ☐ php artisan make:controller TestController -invoke
- ☐ php artisan make:controller TestController --invokable

(01 Pt)

5- Quelle est la commande pour générer une classe Mailable ("contactEmail") ?

- ☐ php artisan make:mail contactEmail
- ☐ php artisan create:mail contactEmail
- ☐ php artisan generate:mail contactEmail
- ☐ php artisan make:mailable contactEmail

6- Comment assigner des alias à un middleware dans Laravel, et où est définie la liste par défaut des alias middleware ?

(02 Pts)

- ☐ En les ajoutant au fichier `app/Http/middleware.php`, qui contient la liste par défaut des alias de middlewares.
- ☐ En les ajoutant au fichier `app/Http/Kernel.php`, qui contient la liste par défaut des alias de middleware
- ☐ En les ajoutant au fichier `app/Http/Kernel.php`, qui contient des entrées pour les middlewares inclus dans Laravel.
- ☐ En les ajoutant au fichier `app/Http/middleware.php`, qui contient des entrées pour les middlewares inclus dans Laravel.

7- Quel sera affiché sur la page Web lorsque le code Blade suivant sera exécuté ?

(02 Pts)

a) `$variable = '<h1>Titre d'article 01</h1>';`

`{{ $variable }}`

☐ `<h1>Titre d'article 01</h1>`

☐ `{{ $variable }}`

☐ Titre d'article 01

☐ `"{{ $variable }}"`



b) \$variable = '<h1>Titre d'article 02</h1>' ;

{{!! \$variable !!}}

☐ <h1>Titre d'article 02</h1>

☐ {{ \$variable }}

☐ Titre d'article 01

☐ {{!! \$variable !!}}



### Pratique : (30 Points)

#### Exercice 01 :

(05 Pts)

Soit le contrôleur `TestController`, le middleware `'TestMiddleware'`.

1. On veut accéder au méthode `index` du `TestController` via la route suivante

(02 Pts)

`Route::get('Test', [TestController::class, 'index']);`

Utiliser cette route pour Appliquer Le middleware `'TestMiddleware'` avec un paramètre de valeur `'admin'`.

.....

.....

.....

2. Dans le contrôleur `TestController` Déclarer un constructeur puis Appliquer

(03 Pts)

- ✓ Un middleware `'midd_2'` à toutes les méthodes qui sont dans le contrôleur, sauf la méthode `store` et `create`.

```
8 class TestController extends Controller
9 {
10
11
12
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20
21 }
22
```



**Exercice 02 :**

Un journal qui souhaite développer un site web pour publier des articles journalistiques dans lesquels les lecteurs interagissent avec l'auteur à l'aide de commentaires.

Les deux tables suivantes font partie de la base de données qui sera utilisée dans ce projet sur Laravel :

✓ **Table : articles**

- Id (entier, clé primaire, auto-incrément)
- Titre (chaîne de caractères)
- Body (texte)
- Catégorie (chaîne de caractères)

✓ **Table : commentaires**

- Id (entier, clé primaire, auto-incrément)
- Pseudo (chaîne de caractères)
- Commentaire (texte)
- Article\_id (entier, clé étrangère faisant référence à la colonne id de la table articles)



Remarque : Nous avons créé les migrations (create\_articles\_table et create\_commentaires\_table) artisan pour ces tables par des commandes.

1. Donner le contenu de la fonction up de chaque migration :✓ **create\_articles\_table**

(02 Pts)

✓ **create\_commentaires\_table**

(02.5 Pts)

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('articles', function (Blueprint $table) {
        // le contenu ici
    });
}
```

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('commentaires', function (Blueprint $table) {
        // le contenu ici
    });
}
```

2- Dans le Model **Article** créer une fonction **commentaires** qui return la relation entre le modèle Article et le modèle Commentaire (02 Pts)

3- Dans le Model **Commentaire** créer une fonction **article** qui return la relation entre le modèle Article et le modèle Commentaire (02 Pts)

Réservé à l'établissement Code : .....



### Exercice 03 :

Dans le Contrôleur `articlesController` créer toutes les fonctions suivantes, et utilisez la validation sur la fonction `update`.

- ✓ `index` Retourne la vue '`articles.index`' avec une liste paginée d'articles avec 10 articles par page (02 Pts)
- ✓ `edit` Retourne la vue '`articles.edit`' avec l'article demandé par ID (02 Pts)
- ✓ `update` pour modifier l'enregistrement du formulaire d'édition d'un article demandé, Rediriger vers le nom de route '`articles`' (02 Pts)
- ✓ `destroy` la suppression d'un article, Rediriger vers le nom de route '`articles`' (02 Pts)

### Remarque :

- ✓ Utilisez la validation sur la fonction `update`.
- ✓ Tous les champs sont obligatoires (pour la validation).

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class articlesController extends Controller
{
```



**Exercice 04 :**

Le fichier **web.php** est responsable de la gestion des requêtes HTTP effectuées sur l'URL, Ce fichier est situé dans le répertoire **routes**, C'est là que vous pouvez définir les routes de l'application pour les requêtes HTTP qui proviennent du Web.

1. Dans le fichier **web.php** Créer les routes resource pour le contrôleur **articlesController** (1 Pt)

2. Dans le fichier **web.php** créer les routes pour les fonctions (update, destroy) de contrôleur **CommentairesController** (1 Pt)



Verbe	URI	Action
GET	/articles	
GET		create
POST		store
GET	/ articles /{id}	
GET		Edit
PUT/PATCH	/ articles /{id}	
DELETE	/ articles /{id}	

Exercice 05 :

(3 Pts)

La vue "articles.create" hérite de la vue "layouts.app", cette vue ("articles.create") a un formulaire dans la section qui appelle "content" pour ajouter un nouvel article.

Remarque : Vous devez utiliser

- ✓ La fonction `route` pour définir l'action de ce formulaire.
- ✓ Le `csrf` à l'intérieur du formulaire.

Donner le code balade de cette vue "articles.create" :

