* **Creation d’un nouveau projet Laravel**
* **Installer laravel :**

**Composer global require laravel/installer**

* **Creer projet laravel methode1:**

**Laravel new myproject**

* **Creer projet laravel methode 2**

**Composer create-project laravel/laravel myproject**

* **Laravel Artisan**

**Laravel artisan est une interface en ligne de commande qui va vous permettre de gérer votre application en lançant des commandes via le terminal.**

1. **Afficher la liste des commandes artisan**

**Php artisan list**

1. **Pour afficher un écran d'aide, faites précéder le nom de la commande de help**

**php artisan help migrate**

1. **Pour créer un modèle**

**php artisan make:model NomDuModel**

1. **Vérifier la version de Laravel installée**

**php artisan -version**

1. **lancer le serveur , lancer le projet Laravel**

**php artisan serve**

1. **voir toutes les routes de votre application**

**artisan route:list**

1. **Exécuter toutes les migrations**

**php artisan migrate**

1. **faire un retour en arrière pour la dernière migration**

**php artisan migrate : rollback**

* **Gestion du routage**
* **Les fichiers de routage par défaut**

**use Illuminate\Support\Facades\Route;**

* + **Routage de base :**

**Route ::get(“/”,function(){**

**Return view(“welcome“)}) ;**

**Route ::get(“/“,“welcome“)**

**Route ::get(“/“,function(){return “Bonjour tous le monde“})**

* + **Routage de controlleur :**

**Route ::get(“/user“,[UserController::class,“methode“])**

* + **Routage nommés :**

**Route ::get(“/“,function(){return “hi“})->name(“home֧)**

* + **Parametres de route**

**Route ::get(“{n}“,function($n){ return “je suis la page“.$n})**

* + **Route match**

**Route ::match([“get“,“post“],function(){return view(“welcome“)})**

* + **Route any**

**Route ::any(“/test“,function(){return view(“welcome“)})**

* **Les Controleurs :**

**Php artisan make :contoller NomController**

**Php artisan serve**

* + **Créer un contrôleur sous Laravel :**

**Namespace App\Http\Controllers ;**

**Use Illuminate\Http\Request ;**

**Class UserController extends Controler**

**{**

**Public function index(){**

**Return view(“welcome“) ;**

**}**

**}**

**Route ::get(“/“,[UserController ::class,“index“])**

* + **Utilisation des paramètres**

**Class UserController extends Controler**

**{**

**Public function get($id){**

**Return “page“.$id ;**

**}**

**}**

**Route ::get(“/get/{id},[UserController ::class,“get“]) ;**

* + **Controlleurs à simple action**

**Cmd :php artisan make :controller HomeController –invokable**

**-invoke :magic method is a special method in PHP classes that allows the object to be invoked as a function.**

**class HomeController extends Controller**

**{**

**Public function \_\_invoke (Request $request)**

**Return “Bienvenue à la page home“ ;**

**}**

**}**

**Route::get('/home', HomeController::class);**

* + **Contrôleur de ressources**

**Cmd : php artisan make :controller HomeController –resource**

**--resouce :create crud methode**

**Called as static fun : Route ::resource(“/“,PostController::class)**

* **Les vues :**
  + **Les structures de controle :**
  + **@if/@elseif/@else,@switch,@isset,@empty**
    - **@if/@elseif/@else**

**$animal = “cheval “ ;**

**@if ( $animal === 'chien' )**

**<p>L'animal est un chien.</p>**

**@elseif ( $animal === chat )**

**<p>L'animal n’est pas un chien.</p>**

**@else <p>L'animal ni un chat ni un chien.</p>**

**@endif**

**<p>L'animal ni un chat ni un chien.</p>**

* + - **@switch**

**@switch($age)**

**@case($age<18)**

**<p>La personne est mineure.</p>**

**@break**

**@case($age>18)**

**<p>La personne est mineure. </p>**

**@break**

**@default**

**<p>Valeur par défaut </p>**

**@endswitch**

* + - **@isset**

**$produit=’costume’ ;**

**@isset($produit)**

**<p>Le produit existe</p>**

**@endisset**

* + - **@empty**

**$produit=““;**

**@empty($produit)**

**<p>Le produit existe</p>**

**@endempty**