



## OBJECTIFS

### Objectifs pédagogiques

Développer une application web à l'aide d'un framework PHP.  
Mise en place de l'architecture Modèle-vue-contrôleur ou MVC  
Utiliser un système de template

### Compétences développées :

- Installation d'un framework PHP
- Utiliser composer
- Configuration de son éditeur de code pour PHP et Laravel
- Associer des URL à des pages/fonctionnalités.
- [cc] Mettre à jour son code
- [cc] Éviter la redondance

### Démarche pédagogique

Découverte et installation du framework Laravel  
Remise du code de la boutique et mise en œuvre dans LARAVEL.

# Introduction

Toutes les étapes sont à réaliser dans le même dossier de projet.

Pour le travail en îlot (itération 3), chacun travaillera dans une branche GIT puis l'îlot fera le choix de celle mergée dans la branche principale après une code review collective.

**⚠ AU MOINS UN COMMIT GIT EST DEMANDÉ À LA FIN DE CHAQUE ÉTAPE**

## À propos des versions de Laravel

*Laravel and its other first-party packages follow [Semantic Versioning](https://semver.org/). Major framework releases are released every six months (~March and ~September), while minor and patch releases may be released as often as every week. Minor and patch releases should never contain breaking changes.*

<https://laravel.com/docs/8.x/releases>

**Note :** Les versions de Laravel depuis 5.5 sont compatibles avec la version 8.

Les tutoriels, forums et documentations sont donc tout à fait applicables en version 8.

En revanche, les versions antérieures ne le sont pas !

## Introduction au Framework

Échange avec la/le formatrice/formateur sur l'intérêt d'un Framework.

# ITÉRATION 1

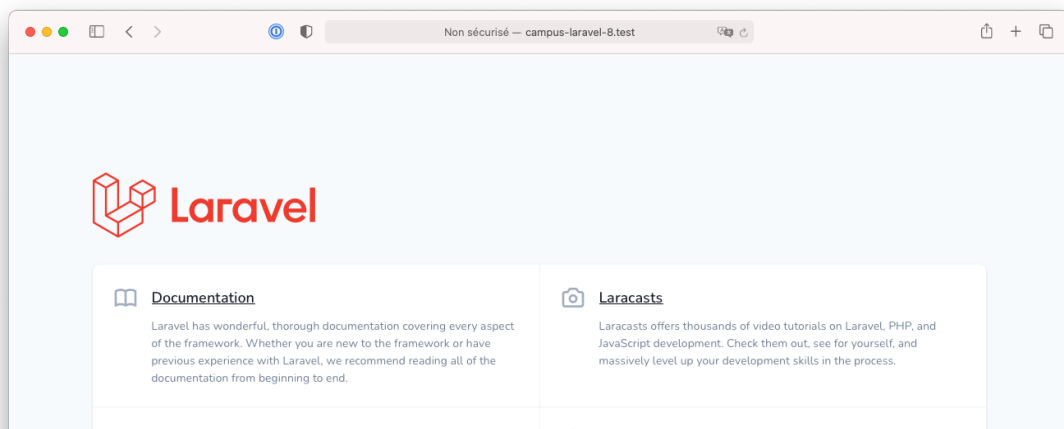
## Installation du Framework et son environnement de développement

### Modalités

- Travail en autonomie
- Production individuelle
- 1 journée

### Livrables

- ❑ Framework fonctionnel (Je suis capable d'afficher la page d'accueil de Laravel)



- ❑ La page d'accueil modifiée pour afficher "Hello World From Laravel v8.X (PHP v7.4.X)"



- ❑ Debugbar installée et fonctionnelle sur votre Laravel



- ❑ Copie d'écran de la debugbar indiquant le fichier "view" utilisé par la page d'accueil.
- ❑ Copie d'écran de la debugbar indiquant la route mise en œuvre.
- ❑ Votre éditeur de code ou IDE avec la coloration syntaxique adaptée au code PHP, plugin spécifique à LARAVEL installé.
- ❑ Le dépôt GitHub est à jour avec la version locale.

## Objectifs

- Installer et faire fonctionner le Framework Laravel "en local" sur votre propre machine.
- Utiliser COMPOSER pour installer les dépendances.
- Comprendre à quoi sert COMPOSER.
- Mettre en oeuvre les outils de développement et debug

## Consignes

### Étape 1 : Installation du Framework

- Installer et faire fonctionner **Laravel version 8 via COMPOSER**
- Créer un dépôt privé sur GitHub et y versionner le projet Laravel
- Modifier la page d'accueil pour afficher le message "Hello World From Laravel v8.X (PHP v7.4.X)" (cf livrables)

**COMMIT GIT attendu à la fin de cette étape**

## Étape 2 : Installation de la debugbar Laravel

Laravel est un framework composé de centaines de fichiers et dépendant de milliers de fichiers, il est primordial de mettre en place des outils de debug pour pouvoir clairement identifier ce qui se passe dans notre code.

Pour HTML et CSS vous utilisez la console de développement de votre navigateur. Pour Laravel vous allez utiliser la "Debugbar" <https://github.com/barryvdh/laravel-debugbar>

- Installer et faire fonctionner la debugbar Laravel

**COMMIT GIT attendu à la fin de cette étape**

## Etape 3 : Configuration de son éditeur de code

Mettre en place les extensions dédiées à Laravel dans votre éditeur de code (voir ressources)

## Compétences

- Installation un framework PHP
- Utiliser composer
- Configuration de son éditeur de code pour PHP et Laravel
- [cc] Mettre à jour son code

## Ressources

- Documentation Laravel
  - <https://laravel.com/docs/8.x>
  - <https://laravel.com/docs/8.x/installation#installation-via-composer>
- Laracasts
  - <https://laracasts.com/series/laravel-6-from-scratch/episodes/2>
  - <https://laracasts.com/series/laravel-6-from-scratch/episodes/5>
- Laravel Debugbar
  - [barryvdh/laravel-debugbar: Laravel Debugbar \(Integrates PHP Debug Bar\)](#)
  - [\[LaraGuide 045\] Installation et présentation de la DebugBar avec Laravel](#)
- Informations sur Composer
  - [Tutoriel vidéo Composer : Composer](#) (les 7 premières minutes)
  - [Tutoriel PHP: Composer & PHPDocumentor](#)
- Pour aller plus loin avec Composer
  - [PHP Tutorial: Getting Started with Composer](#)
- IDE et plugin
  - [PhpStorm: The Lightning-Smart IDE for PHP Programming by JetBrains](#) (licence étudiant avec votre email campus)
  - [Visual Studio Code - Code Editing. Redefined](#)
  - [Visual Studio Code for PHP Developers](#)

- 
- [PhpStorm plugins for Laravel – Episode 3 | PhpStorm Tips with Christoph Rumpel](#)
  - [Atom Code Editor : laravel package & blade package](#)

# ITÉRATION 2

## Mise en place de l'arborescence dans LARAVEL

### Modalités

- Travail en autonomie
- Production individuelle
- 1 journée

### Livrables

- ☐ Pages de votre application accessibles via des URL individuelles
  - ☐ Home page
  - ☐ Liste des produits
  - ☐ Détails produit
  - ☐ Panier
- ☐ Mémo de la hiérarchie d'un projet LARAVEL

### Objectifs

- Mettre en place le **routage** de vos pages dans LARAVEL
- Créer un CONTROLLER
- Afficher une VIEW

### Consignes

#### Étape 1 : Mise en place des URL dans votre fichier de route

Cette première étape a pour but de vous faire découvrir le routage dans LARAVEL.

Il ne s'agit pas pour le moment d'afficher des pages fonctionnelles mais uniquement un **texte** simple permettant de valider le bon fonctionnement de vos URLs

**VOUS N'UTILISEREZ PAS LES VIEWS**

Editer le fichier des routes pour ajouter une route pour chaque page du site :

PAGE	URL	Texte à afficher sur la page
home page	/	Home page
Liste des produits	/product	Liste des produits
détails produit	/product/{id}	Fiche du produit {id}
Panier	/cart	Panier

{id} est un paramètre dynamique de l'URL ,  
exemple : /product/1 doit afficher "Fiche du produit 1"

**COMMIT GIT attendu à la fin de cette étape**

 faire valider cette étape par un formateur

## Étape 2 : Création de CONTROLLER

Créer 3 controller :

- HomeController
- ProductController
- CartController

et adapter le code de votre site de façon à répondre au fonctionnement suivant :

**VOUS N'UTILISEREZ PAS LES VIEWS**

PAGE	URL	Controller	Texte à afficher sur la page
home page	/	HomeController	Home page
Liste des produits	/product	ProductController	Liste des produits
détails produit	/product/{id}	ProductController	Fiche du produit {id}
Panier	/cart	CartController	Panier

{id} est un paramètre dynamique de l'URL ,  
exemple : /product/1 doit afficher "Fiche du produit 1"

**COMMIT GIT attendu à la fin de cette étape**

 faire valider cette étape par un formateur



## Étape 3 : Création des VIEWS

Créer 4 view

- homepage.blade.php
- product-list.blade.php
- product-details.blade.php
- cart.blade.php

et faire en sorte de maintenir le fonctionnement de l'étape 2 (affichage de textes)

PAGE	URL	Controller	views	Texte à afficher sur la page
home page	/	HomeController	homepage.blade.php	Home page
Liste des produits	/product	ProductController	product-list.blade.php	Liste des produits
détails produit	/product/{id}	ProductController	product-details.blade.php	Fiche du produit {id}
Panier	/cart	CartController	cart.blade.php	Panier

{id} est un paramètre dynamique de l'URL ,

exemple : /product/1 doit afficher "Fiche du produit 1"

**COMMIT GIT attendu à la fin de cette étape**

 faire valider cette étape par un formateur

## Étape 4 : Organisation des fichiers dans un projet LARAVEL

Cette étape vous a donné l'occasion de naviguer dans les fichiers de LARAVEL.

Rédiger un mémo sur la hiérarchie de votre site (répertoires / sous répertoires / fichiers) dans LARAVEL

```
▼ campus-laravel-8 ~/Sites/campus-laravel-8
  > app
  > bootstrap
  > config
  > database
  > public
  > resources
  > routes
  > storage
  > tests
  > vendor
```

 faire valider cette étape par un formateur

## Compétences

- Associer des URL à des pages/fonctionnalités.
- [cc] Programmer avec des variables
- [cc] Mettre à jour son code

## Ressources

- Documentation Laravel
  - <https://laravel.com/docs/8.x/routing>
  - <https://laravel.com/docs/8.x/controllers>
  - <https://laravel.com/docs/8.x/views>
- Laracasts
  - <https://laracasts.com/series/laravel-6-from-scratch/episodes/5>
  - <https://laracasts.com/series/laravel-6-from-scratch/episodes/6>

## ITÉRATION 3

# Intégration de l'HTML / CSS + préparation Frontoffice.

### Modalités

- Travail en îlot en autonomie avec partage des tâches
- Production de groupe.
- 1 journée

### Livrables

- ☐ Maquette de votre site intégré dans LARAVEL
- ☐ Vos CSS inclus dans LARAVEL
- ☐ Navigation fonctionnelle entre les différentes pages

### Objectifs

- Nous souhaitons pouvoir naviguer entre les pages de votre application
- Nous souhaitons factoriser notre code afin d'éviter les doublons et d'en faciliter la maintenance .

### Consignes

#### Étape 1 : Intégration HTML

- Intégrer le code HTML de votre projet dans vos pages LARAVEL.
- Faites en sorte que les ressources associées à vos pages (fichiers CSS) soient bien chargées.

**Chaque stagiaire devra effectuer l'intégration d'au moins une VIEW et de sa route associée**  
**COMMIT GIT attendu à la fin de cette étape**

#### Étape 2 : Suppression des éléments dupliqués du code

- Créez et intégrez des éléments réutilisables dans vos pages pour ne pas répéter du code entre les pages (En-tête , Pied de page, Menus, ...)

**Chaque stagiaire devra effectuer l'intégration d'au moins une VIEW et de sa route associée**  
**COMMIT GIT attendu à la fin de cette étape**

Il n'est pas demandé d'utiliser SASS / LARAVEL MIX

## Compétences

- [cc] Éviter la redondance

## Ressources

- Code HTML/CSS de votre boutique
- <https://laravel.com/docs/8.x/blade>
- <https://laravel.com/docs/8.x/helpers>
- <https://www.grafikart.fr/formations/laravel/view-template-blade>