

# Laravel 12 – Développement Web PHP

4 jours (28 heures)

Perfectionnement

Présentiel ou classe à distance

à partir de

**2 540 € HT** par personne (TVA en vigueur) Repas inclus en présentiel

Contenu mis à jour le 25/09/2026. Document téléchargé le 04/10/2026.

## Objectifs

A l'issue de cette formation, vous serez capable de :

- Décrire et mettre en oeuvre le framework Laravel
- Implémenter toutes ses fonctionnalités.

Les compétences visées par cette formation font l'objet d'une évaluation.

## À qui s'adresse cette formation ?

Développeurs, architectes, chefs de projets techniques.

### Prérequis

Avoir la connaissance nécessaire du langage PHP ainsi que de ses utilisations dans le monde objet.

## Programme

### 1 Pendant votre formation

## Laravel 12 – Développement Web PHP

Jour 1 – Matin

## Architecture de Laravel

- Présentation de Laravel
- Les services Provider
- Les services Container
- Les façades
- Cycle de vie d'une requête en Laravel
- Structure type d'une application

## Installation

- Exigences système
- Installation
- Configuration du serveur Web
- Structure du projet

### *Exemples de travaux pratiques (à titre indicatif)*

- *Installation de Laravel*
- *Paramétrage initial et conception d'une petite application*

## Le routage

- Routage basique
- Paramètres des routes
- Les routes nommées, les routes sécurisées
- Les contraintes appliquées aux paramètres
- Préfixes de routes
- Routage de domaine

### *Exemples de travaux pratiques (à titre indicatif)*

- *Présentation de l'application globale à développer*
- *Création des premières routes qui seront traitées par les contrôleurs à venir*

Jour 1 – Après-midi

## Les contrôleurs

- L'architecture MVC
- Place du contrôleur dans l'architecture MVC
- Introduction aux contrôleurs
- Un contrôleur basique
- Les contrôleurs implicites
- Les contrôleurs de ressource REST

### *Exemples de travaux pratiques (à titre indicatif)*

- *Reprise de l'application en cours de développement*
- *Création de plusieurs contrôleurs de différents types, notamment REST*

## Blade – syntaxe et structures

- Affichage des données et maîtriser l'échappement
- Itération et conditionnement avec les directives Blade
- Gestion des états utilisateur et la session
- Quand utiliser du HTML brut et les risques XSS

*Exemple de travaux pratiques (à titre indicatif)*

- Afficher une liste d'articles depuis un contrôleur dans une vue Blade

## Les templates

- Création des templates
- Structures de contrôle
- Templates et héritage de templates

*Exemple de travaux pratiques (à titre indicatif)*

- Utilisation de templates pour l'application en cours de développement

## Les réponses

- Avec les vues ou avec JSON
- Les redirections
- Les réponses customisées
- Les messages flash

*Exemples de travaux pratiques (à titre indicatif)*

- Faire varier la réponse d'un même endpoint selon le format demandé (HTML, JSON)
- Ajout de notifications flash dans l'application en cours de développement

Jour 2 – Matin

## Génération d'URL

- L'URL courante
- La génération des URL
- Les raccourcis d'URL

## Les middlewares

- Le rôle d'un middleware dans le pipeline de requête
- Attacher un middleware globalement ou à une route
- Les middlewares natifs – auth, throttle, signed, verified

*Exemple de travaux pratiques (à titre indicatif)*

- Création d'un middleware Laravel avec Artisan

## Service Container et Service Providers

- Injection de dépendances par résolution automatique

- Déclaration d'un binding simple ou en singleton
- Distinguer phase register et phase boot d'un Service Provider

*Exemple de travaux pratiques (à titre indicatif)*

- Créer un service applicatif simple, l'enregistrer via un Provider et l'injecter dans un contrôleur

Jour 2 – Après-midi

## La base de données

- Choix du SGBD
- Configurer la connexion via .env
- Gérer les connexions multiples

## Migrations

- Créer une migration via Artisan
- Définir un schéma de table avec colonnes, contraintes et index
- Gérer la rétroversion d'une migration
- Exécuter, annuler ou réinitialiser les migrations

*Exemples de travaux pratiques (à titre indicatif)*

- Configurer le projet pour SQLite en local
- Modéliser les tables users, posts, comments avec leurs migrations et relations

## Query Builder

- Enchaîner des conditions de filtrage
- Ordonner, limiter et paginer un résultat
- Joindre des tables et agréger des données

*Exemple de travaux pratiques (à titre indicatif)*

- Ecrire un ensemble de requêtes Query Builder pour produire un rapport (top 10 articles, répartition par auteur, etc...)

Jour 3 – Matin

## Modèles Eloquent

- Créer un modèle Eloquent et générer ses artefacts liés en une commande Artisan
- Comprendre les conventions de nommage et de timestamps
- Protéger les attributs contre le mass assignment

*Exemple de travaux pratiques (à titre indicatif)*

- Créer les modèles Post et Comment avec leurs migrations, factories et conventions

## Casts et attributs

- Typer automatiquement les colonnes avec les casts (notamment enum)
- Définir des accessors et mutators modernes via Attribute::make()

- Exploiter les attributs virtuels calculés

*Exemple de travaux pratiques (à titre indicatif)*

- Ajouter un attribut calculé `excerpt` sur `Post` et un `cast enum` sur le statut

### Les relations

- Définir les relations 1:1, 1:N et N:N
- Exploiter les tables pivot avec attributs supplémentaires
- Mettre en oeuvre des relations polymorphes

*Exemple de travaux pratiques (à titre indicatif)*

- Implémenter `User hasMany Post`, `Post hasMany Comment` et `Post belongsToMany Tag` puis exploiter ces relations

### Requêtes Eloquent

- Récupérer une entité ou une collection
- Filtrer avec contraintes sur les relations
- Définir des scopes locaux pour réutiliser des filtres métier
- Éviter le problème N+1 grâce à l'eager loading

*Exemple de travaux pratiques (à titre indicatif)*

- Implémenter une recherche d'articles publiés avec leurs auteurs et un nombre de commentaires, sans N+1

Jour 3 – Après-midi

### Formulaires Blade et CSRF

- Générer un formulaire HTML avec jeton CSRF
- Afficher les erreurs et conserver les saisies en cas d'échec
- Styliser les états valide / invalide

*Exemple de travaux pratiques (à titre indicatif)*

- Créer le formulaire de création d'article avec persistance des saisies en cas d'erreur

### FormRequest et validation

- Créer une `FormRequest` dédiée par cas
- Déclarer les règles métier et utiliser les règles natives
- Personnaliser les messages et noms d'attributs
- Écrire une règle de validation custom

*Exemple de travaux pratiques (à titre indicatif)*

- Refondre la validation d'un formulaire complexe (création d'article avec image et tags) dans une `FormRequest`

### Authentification avec Breeze

- Installer un starter kit avec choix Blade, Livewire ou Inertia
- Comprendre les routes d'authentification générées
- Personnaliser vues, contrôleurs et redirections

*Exemple de travaux pratiques (à titre indicatif)*

- Installer Breeze (Blade), explorer les routes et personnaliser la page de connexion

**Autorisation – Gates et Politiques**

- Déclarer un Gate pour une autorisation transverse
- Créer une Policy liée à un modèle
- Protéger des actions depuis Blade, un contrôleur ou un middleware

*Exemple de travaux pratiques (à titre indicatif)*

- Restreindre la modification et la suppression d'un article à son auteur via une Policy

**Sessions**

- Comprendre le rôle de la session côté serveur
- Choisir un driver de session adapté au contexte
- Stocker, lire et oublier des données de session
- Protéger la session – régénération de l'identifiant, expiration, cookies sécurisés

Jour 4 – Matin

**Artisan CLI**

- Explorer les commandes essentielles du quotidien
- Créer une commande Artisan custom
- Planifier des tâches récurrentes avec le Scheduler

*Exemple de travaux pratiques (à titre indicatif)*

- Créer une commande `app:cleanup-expired` exécutée quotidiennement

**Les mails**

- Créer une Mailable
- Composer un mail en Markdown ou via une vue, avec pièces jointes
- Configurer un driver SMTP

*Exemple de travaux pratiques (à titre indicatif)*

- Envoyer un mail de bienvenue après l'inscription d'un utilisateur

**Queues et Jobs**

- Comprendre pourquoi déporter du travail en arrière-plan
- Créer un Job persistable
- Configurer un driver de queue (database, Redis)
- Lancer un worker et superviser les échecs

*Exemple de travaux pratiques (à titre indicatif)*

- Déporter l'envoi du mail de bienvenue dans un Job persisté en base et exécuté par un worker

**Events et Listeners**

- Déclencher un événement applicatif
- Y réagir via un Listener classe ou closure
- Exploiter les Observers Eloquent sur le cycle de vie d'un modèle

*Exemple de travaux pratiques (à titre indicatif)*

- Déclencher un événement à la création d'un article et écouter pour notifier les abonnés

## Notifications

- Créer une Notification
- Choisir un canal de diffusion (mail, base de données, broadcast)
- Afficher les notifications stockées en base à l'utilisateur

*Exemple de travaux pratiques (à titre indicatif)*

- Notifier l'auteur d'un article quand un commentaire est posté, en mail et en database

## Cache

- Choisir un driver de cache adapté à l'environnement
- Mettre en cache et invalider intelligemment
- Exploiter le tagging pour invalider par groupe

*Exemple de travaux pratiques (à titre indicatif)*

- Mettre en cache la liste des articles populaires pendant 5 minutes et observer l'effet sur les performances

Jour 4 – Après-midi

## Tests avec Pest

- Découvrir Pest, runner de tests par défaut depuis Laravel 11
- Ecrire un test unitaire et un test fonctionnel
- Tester une route HTTP avec assertions sur la réponse
- Isoler la base entre les tests grâce aux factories

*Exemple de travaux pratiques (à titre indicatif)*

- Ecrire des tests Pest sur le contrôleur de ressource d'articles, avec authentification

## API Resources et Sanctum

- Transformer une réponse JSON via une API Resource
- Composer des collections et des relations imbriquées
- Protéger une API avec Sanctum (tokens personnels ou SPA cookie)

*Exemple de travaux pratiques (à titre indicatif)*

- Exposer les articles en API JSON via Resource, sécurisée par Sanctum (token)

## Outillage qualité

- Formater le code avec Laravel Pint
- Déboguer avec Laravel Telescope

- Monitorer la production avec Laravel Pulse
- Positionner Laravel Octane pour la performance

*Exemple de travaux pratiques (à titre indicatif)*

- Intégrer Pint à la CI, installer Telescope en environnement local

### Ecosystème Laravel 13 (panorama)

- Comprendre Livewire 3 et son intégration aux starter kits
- Positionner Inertia.js comme alternative SPA
- Découvrir Filament pour les interfaces d'administration
- Situer Volt et Folio dans la stack moderne

*Exemple de travaux pratiques (à titre indicatif)*

- Créer un mini-composant Livewire de recherche d'articles intégré dans le projet

### Laravel et IA

- Situer les assistants disponibles
- Formuler un prompt qui précise la version Laravel et les conventions attendues
- Repérer les pièges spécifiques – versions Laravel mélangées dans les exemples, API dépréciées, conventions manquées
- Vérifier systématiquement le code généré par exécution Artisan et tests Pest

## Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

[Nous contacter pour en savoir plus →](#)

## Modalités pédagogiques

Formation délivrée en présentiel ou distanciel\* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode\*\* démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2I, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

\* nous consulter pour la faisabilité en distanciel

\*\* ratio variable selon le cours suivi

### Accessibilité aux personnes en situation de handicap

## Détails de la session

Du 3 novembre 2026 au 6 novembre 2026 (4 jours / 28 heures)

Lyon

Du 3 novembre 2026 au 6 novembre 2026 Lyon

- 3 novembre 2026 ..... 9h00 – 15h30
- 4 novembre 2026 ..... 9h00 – 15h30
- 5 novembre 2026 ..... 9h00 – 15h30
- 6 novembre 2026 ..... 9h00 – 15h30