

Framework PHP - B3

Durée: 35h (5 jours)

Version : 1

Auteur: Paul Schuhmacher

Ce que vous saurez faire à la fin du module

- **Maîtriser le PHP Moderne (8+):**
 - Utiliser les features modernes de PHP (namespace, autoloading);
 - Utiliser et connaître les dernières nouveautés de PHP 8;
 - Utiliser le gestionnaire de paquets `Composer` dans vos projets;
 - Utiliser des composants dans vos projets (Symfony ou autre) avec Packagist;
 - Écrire une codebase PHP **maintenable**: codebase *linté* et *formaté* en suivant des standards avec Git, PHP_CodeSniffer, PHPStan et les standards PSR-*;
- **Maîtriser l'architecture MVC** et créer une application web suivant cette architecture;
- **Utiliser des frameworks à bon escient** (ne pas réinventer la roue, ou uniquement pour des bonnes raisons);
- Avoir les *bases nécessaires* pour **développer des applications web avec le framework Symfony**;
- **Mettre en production** une application web;

Programme

Rappels POO, PHP, PHP Moderne (8+) et pattern MVC

- PHP moderne: Namespaces, *autoloading*, Composer et Packagist;
- Maintenance du projet *Linting* (Analyse Statique, règles de composition) et *Formating* avec phpCodeSniffer, PHPStan et les standards PSR;
- Rappels Programmation Orientée Objet en PHP;
- L'architecture (ou design pattern) *Model View Controller* (MVC);
- Implémentation du pattern MVC en *vanilla* PHP;

Symfony : Présentation, setup et premiers pas

- Présentation de Symfony;
- Installation de Symfony;
- Structure d'un projet Symfony;
- Configuration;
- Premiers pas;

L'architecture du framework et pattern MVC

- L'architecture (ou design pattern) *Model View Controller* (MVC) avec Symfony;
- L'architecture Symfony, sous le capot;
- L'*autoloading* de Symfony (PSR-4);
- La console;
- Debug: profiler Symfony et `dump()` ;

Routage et Contrôleurs

- Routage dans Symfony
- Définir des routes
- Le contrôleur
- Configurer le path
- Règles, services, URL, gestion des erreurs
- TP

L'injection de dépendances et configurations

- Les environnements
- Utilisation de la console symfony
- L'injection de dépendances et design pattern IoC: le service Container de Symfony
- L'*autowiring* (injection de dépendances automatique)
- Exemple: Injecter un service dans un service (la boucle est bouclée)
- Le dossier public pour les *assets*
- TP

Templating avec Twig

- présentation du moteur de template Twig
- pourquoi Twig au lieu de PHP ?
- Concepts de Twig: héritage et blocks
- Langage Twig, syntaxe
- Manipulation des variables et structures de contrôle
- Sécurité: l'échappement avec Twig
- Des templates concis, lisibles et performants: introduction aux filtres et aux fonctions
- Accéder aux assets
- Exemple: installation du framework CSS Bootstrap dans une application Symfony
- TP

L'accès à la base de données avec l'ORM Doctrine

- Mise en place de la base de données avec Docker
- Accéder à la base de données avec le client CLI `psql` et un client graphique (adminer)
- L'ORM Doctrine, de l'objet au relationnel et inversement
- Architecture et Doctrine
- Notion d'*entité*
- Configuration des entités via les attributs PHP
- Mise à jour du schéma et gestion des migrations
- TP

Les Formulaires

- Créer un formulaire dans Symfony
- Rendre un formulaire dans un Template
- Traiter et valider les données d'un formulaire
- TP

Éléments de sécurité

À venir...

Mise en production

- Configuration de PHP;
- Performances;
- Journalisation;

À venir...

Évaluation

- QCM début séance 3 (Coeff 1);
- Développement d'une application web Symfony par groupe de 2 (Coeff 4);

Références utiles

- [Dépôt du cours](#), mis à jour au cours du module;
- [Dépôt POO et Design Patterns avec PHP](#)
- [Dépôt veille PHP](#)
- [Dépôt starter Symfony](#), à *mettre à jour*;
- [Documentation officielle de Symfony](#)
- *Modern PHP*, de Josh Lockhart publié chez O'REILLY (2015)
- [Symfony 5, Développez des sites web PHP structurés et performants](#), d'Etienne langlet, publié chez ENI Éditions (2022). Toujours pertinent.