
Formation Symfony4

« Les bases du développeur PHP »

De Novice à Ninja



3 jours (21 heures)



Intra-entreprise : nous nous déplaçons chez vous pour former votre équipe (3 ~ 8 participants)
4 200 € HT + frais (déplacement, hébergement, etc.)



Inter-entreprises : inscrivez-vous aux sessions que nous organisons à Nantes ou ailleurs !
1 490 € HT par personne



Renseignez-vous auprès de votre OPCA pour une **prise en charge partielle ou totale du coût de la formation**. En cas de subrogation, pas d'avance de trésorerie, KNP Labs peut facturer directement votre OPCA !

02 85 52 60 25
hello@knplabs.com

Objectifs de la formation

Faire acquérir aux participants les connaissances des mécanismes de base du framework PHP Symfony4 avec l'ORM Doctrine 2.*.

Description de la formation

- Le contenu est actualisé régulièrement afin de toujours couvrir la **dernière version de Symfony4**.
- Le parcours pédagogique se présente sous la forme d'un projet d'application internet sur le thème de la pâtisserie, réalisé par chaque participant. Des travaux pratiques ont lieu tout au long de la formation, qui mènent progressivement les participants à corriger leurs erreurs.
- L'accent est mis sur les **bonnes pratiques**.

Programme

- 1. Présentation**
 - 1.1 Pourquoi un framework
 - 1.2 Apports de Symfony
 - 1.3 Communauté
- 2. Installation & configuration**
 - 2.1 Composer
 - 2.2 Installer Symfony avec composer create-project
 - 2.3 Symfony Flex
 - 2.4 Symfony project architecture
 - Architecture de Symfony3
 - Architecture de Symfony4
- 3. Fichier de configuration**
 - 3.1 Les possibilités de configuration de langage : yml / xml / annotation, et pourquoi?
 - 3.2 L'environnement Symfony
 - 3.3 Recommandations Symfony & ce qui sera utilisé lors de la formation
- 4. Créer une première action**
 - 4.1 Créer un contrôleur qui affiche une chaîne de caractères
 - 4.2 Créer le routing
- 5. Autoloading**
 - 5.1 Composer autoloader
 - 5.2 PSR-4
- 6. Afficher la liste des gâteaux**
 - 6.1 Utiliser PDO pour récupérer les gâteaux
 - 6.2 Les afficher simplement
- 7. Twig**
 - 7.1 Présentation des instructions de base
 - 7.2 Intégration de Symfony
 - Montrer la méthode de contrôleur render
 - 7.3 Conventions de nommage et emplacement de templates
 - 7.4 Créer un premier template pour lister les gâteaux
 - 7.5 Rendre ce template depuis le contrôleur
- 8. Assets (ajout de css)**
 - 8.1 Que sont les assets?
 - 8.2 Utiliser webpack-encore pour compiler les assets
- 9. Debugging**
 - 9.1 Debug toolbar
 - 9.2 dump()
- 10. Configuration de Symfony**
 - 10.1 Paramètre
 - Paramètres fournis par Symfony (ex : `kernel.root_dir`)
 - 10.2 Comment récupérer un paramètre depuis un contrôleur
 - 10.3 Utiliser les paramètres Symfony pour configurer PDO depuis le contrôleur
- 11. Paramètres de routing**
 - 11.1 Comment définir des paramètres de routing
 - 11.2 Comment récupérer ces paramètres

depuis le contrôleur

11.3 Créer une route `display cake` en utilisant un paramètre de routing

12. Twigs paths & urls

12.1 Comment générer des paths & urls depuis un template twig

12.2 Ajouter des liens dans les templates du `toyproject`

13. Refactoring du contrôleur

13.1 Créer un singleton pour configurer PDO à l'intérieur du contrôleur

14. Twig : héritage & blocks

14.1 Comment étendre un template

14.2 Comment définir des blocks

14.3 Créer un template de layout

14.4 Étendre ce layout

15. HTTP status codes

15.1 Qu'est-ce qu'un status code

15.2 Comment configurer le response status code (renvoyer une exception http)

15.3 Renvoyer un exception quand le gâteau n'est pas trouvé

15.4 Détailler les messages d'erreur dans le profiler

16. Créer un gâteau (1ère Partie)

16.1 Créer contrôleur, template et route

16.2 Erreur calculée sur le navigateur:

"le gateau avec l'id `create` n'existe pas"

17. Affiner le routing

17.1 Ajouter des conditions (requirements)

18. Créer un gâteau (2ème Partie)

18.1 utiliser `$_POST` pour récupérer les données soumises

18.2 insérer des données manuellement avec PDO

18.3 Ajouter une redirection vers la liste des gateaux quand le gâteau a été créé

18.4 Quelques explications sur `RedirectResponse`

19. Requête Symfony

19.1 Afficher des paramètres de la requête HTTP disponibles (explication de l'objet `request`)

19.2 Comment avoir la requête dans le contrôleur

19.3 Retirer les usages de `$_POST` et les remplacer par la requête

20. Session (pour ajouter un message flash)

20.1 Décrire le session bag

20.2 Comment ajouter des messages flash dans le session bag

20.3 Utiliser le raccourci contrôleur `addFlash()`

20.4 Afficher des messages flash depuis un template twig

21. Formulaires symfony (1ère partie)

21.1 Comment les `FormType` sont composés

21.2 Introduction au `FormBuilder`

21.3 Créer un `FormType` de création de gateaux avec des champs simples

21.4 Comment instancier le formulaire depuis un contrôleur (`createForm()`)

22. Rendre un formulaire symfony

22.1 Fonctions twig pour formulaires

- `form_start`

- `form_widget` & `form_row`

- `form_end`

22.2 Rendre notre `FormType` de création de gateaux

23. Formulaires symfony (2ème partie)

23.1 Passer la requête au formulaire

23.2 Récupérer les données du formulaire

23.3 Utiliser les données du formulaire pour les sauvegarder

24. Validation symfony

24.1 Présentation

24.2 Appliquer des règles de validation sur entités et champs de formulaires

24.3 Utiliser la méthode `isValid()`

24.4 Ajouter de la validation à notre formulaire

25. Doctrine

- 25.1 Le gestionnaire d'entités (entity manager)
- 25.2 Les ORM : un peu de théorie
- 25.3 Qu'est-ce qu'une entité
- 25.4 Comment mapper une entité (ym1)
 - configurer doctrine bundle pour le mapping yml
- 25.5 Configurer la connexion avec la BDD
- 25.6 Ecrire dans la BDD, persist() & flush()
- 25.7 Créer une entité gâteau
 - Classe, fichier de mapping
- 25.8 Dans le contrôleur
 - Créer une entité gâteau à partir des données du formulaire
 - Enregistrer cette entité avec le gestionnaire d'entités
 - Remplacer le PDO
- 25.9 Quelques mots sur DQL
- 25.10 Migration de BDD (DoctrineMigration)

26. Formulaires symfony (3ème partie)

- 26.1 Mapper les données du formulaire directement sur l'entité
 - utilisation de la propriété data_class

27. Sécurité

- 27.1 user providers
- 27.2 password encoders
- 27.3 firewall
- 27.4 access_control (protéger certaines URL derrière un firewall)

- 27.5 Vérification des droits (is_granted / voter)
- 27.6 hiérarchie des rôles utilisateur

28. Internationalisation

- 28.1 Mettre en place les traductions
- 28.2 Les utiliser dans twig
- 28.3 Utiliser des placeholders dans les traductions
- 28.4 Traduction d'entités

29. Logs

- 29.1 Monolog

30. Swiftmailer**31. Injection de dépendances**

- 31.1 Déclaration de services
- 31.2 Présentation des events du kernel

32. Commandes symfony

- 31.1 Composant console

33. Pour aller plus loin

- 34.1 Behat
- 34.2 Phpspec
- 34.3 FosJsRouting
- 34.4 Event Listener / Subscriber
- 34.5 Dispatching d'événements
- 34.6 API REST / Sérialisation
- 34.7 Bundles utiles recommandés par KNP Labs

Prérequis :

Afin de pouvoir profiter pleinement de la formation, les participants devront impérativement avoir une expérience professionnelle préalable dans les domaines suivants :

- développement PHP5 orienté objet
- utilisation d'une base de données relationnelle

Les participants devront avoir un ordinateur avec les droits admin, leur permettant de faire les exercices proposés tout au long de la formation, avec un environnement LAMP comprenant de préférence les éléments suivant :

- Une connexion internet libre

Organisme de formation n° 52 44 06190 44

- Un bon éditeur PHP (sublime text, php storm ...)
- Un shell UNIX
- PHP 7.1 minimum installé (une des versions les plus récentes si possible)
- Git installé (utile pour des dépendances de composer)
- Un serveur mysql par personne.

Il est conseillé de permettre aux participants d'utiliser un éditeur de texte avec lequel ils se sentent le plus à l'aise afin de se concentrer au maximum sur la formation. **Il n'est pas nécessaire d'installer Symfony, cette partie sera couverte pendant la formation.**

Plus généralement, si la formation se déroule dans les locaux du client, il est demandé qu'une **salle** permettant d'accueillir l'ensemble des participants ainsi que le formateur soit réservée pendant l'intégralité de la durée de la formation. Un moyen de présentation écran (**projecteur** ou téléviseur plat de grande taille) devra être présent dans cette salle afin de permettre aux participants de visualiser l'écran du formateur. Un paperboard ou tableau effaçable et une connexion internet seraient également appréciés.

Matériel pédagogique fourni

Le formateur distribuera les supports de la formation en format électronique à la fin de la formation.

D'autre part, le travail pendant ces 3 jours donnera lieu à la création d'une application web versionnée étape par étape. Chaque participant est invité à garder sur sa machine le code du projet de formation, qu'il aura rédigé lui-même avec l'aide du formateur, car ce code pourra servir de base technique pour de futurs projets.

Formation Symfony4

Modules Complémentaires

Approfondissez vos connaissances sur des aspects très précis du framework en ajoutant à la formation de base un ou plusieurs modules complémentaires.

Faites participer votre équipe au choix des modules, pour assurer le succès de vos projets en cours et à venir !

Sécurité

Durée : 1 jour

Tarif intra-entreprise : 1 400 € + frais (max 8 participants)

Tarif inter-entreprises : 500 € par personne

- Personnaliser les méthodes d'authentification
 - Création d'un firewall listener
 - Création d'un fournisseur d'authentification
 - Création de factory pour le security bundle
- Utiliser des voters pour prendre les décisions

Internationalisation

Durée : 0,5 jour

Tarif intra-entreprise : 700 € + frais (max 8 participants)

Tarif inter-entreprises : 250 € par personne

- A propos du translator
- Mettre en place le translator
- Détecter les utilisateurs locaux

Advanced Service Container

Durée : 1 jour

Tarif intra-entreprise : 1 400 € + frais (max 8 participants)

Tarif inter-entreprises : 500 € par personne

- Pourquoi utiliser un DIC?
- Créer une extension de container
- Rendre une extension de container configurable

- Jouer avec les tags & compiler passes

Tester avec Behat

Durée : 1 jour

Tarif intra-entreprise : 1 400 € + frais (max 8 participants)

Tarif inter-entreprises : 500 € par personne

- Qu'est qu'un BDD? ScenarioBDD?
- Spécifier une fonctionnalité
- Mettre en place Behat
- S'appuyer sur Behat pour l'implémentation

Organiser et optimiser les assets avec Assetic

Durée : 0,5 jour

Tarif intra-entreprise : 700 € + frais (max 8 participants)

Tarif inter-entreprises : 250 € par personne

- A propos d'Assetic
- Mettre en place assetic
- Compresser/compiler des assets

Doctrine avancée

Durée : 1 jour

Tarif intra-entreprise : 1 400 € + frais (max 8 participants)

Tarif inter-entreprises : 500 € par personne

- Utiliser les event listeners
- Utiliser les native queries
- Créer des fonctions DQL personnalisées
- Utiliser le result cache

Formulaires avancés

Durée : 1 jour

Tarif intra-entreprise : 1 400 € + frais (max 8 participants)

Tarif inter-entreprises : 500 € par personne

- Anatomie d'un form type
- Cas pratiques
- Utiliser les listeners & transformers
- Comportements génériques utilisant les type extensions