

## Formation API Platform : Fondamentaux

<b>Durée :</b>	5 jours
<b>Public :</b>	Développeur PHP Symfony confirmé
<b>Pré-requis :</b>	Bonne connaissance en PHP Symfony
<b>Objectifs :</b>	Apprendre à mettre en place une API à l'aide d'API Platform
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PHP102060-F
<b>Note de satisfaction des participants:</b>	Pas de données disponibles

### Introduction

Rappel sur Symfony et les API  
Découvrir API Platform  
Connaitre les avantages et inconvénients

#### **Atelier : Installer API Platform**

### Les bases d'API Platform

Créer une entité  
Exposer une entité  
Lier API Platform à une base de données  
Découvrir la norme de description OpenAPI  
Documenter notre API à travers Hydra, SwaggerUI ou ReDoc  
Choisir le format d'affichage des données (html, json, json+ld)

#### **Atelier : Manipuler des jeux de données à l'aide de SwaggerUI et Postman**

### Les opérations

Personnaliser le CRUD à l'aide des opérations  
Renommer une entité  
Comprendre le fonctionnement des urls et des formats  
Debugger les routes de l'api : la commande debug:router  
Redéfinir les urls d'une entité

#### **Atelier : Personnaliser le crud d'une entité**

### Limitier l'exposition des données

Utiliser le composant de sérialisation de Symfony  
Créer des groupes de sérialisation  
Limiter l'exposition des propriétés en lecture : `normalizationContext`  
Personnaliser l'affichage d'une propriété depuis les accesseurs  
Contrôler les groupes d'écriture : `denormalizationContext`

### **Atelier : Définir les propriétés à afficher et à insérer**

## **Utiliser les règles de validation**

Utiliser le composant de validation de Symfony  
Attribuer des contraintes sur les propriétés  
Créer des groupes de validation  
Associer un groupe de validation à une opération : `validationContext`

### **Atelier : Valider les données lors d'une insertion ou une modification**

## **Pagination et Filtres**

Afficher et modifier la configuration d'api platform  
Obtenir les informations de pagination  
Activer et exposer la pagination dans l'url  
Créer des filtres de données avec l'attribut `ApiFilter`  
Utiliser les filtres par défaut (`DateFilter`, `SearchFilter`, `RangeFilter`, etc)  
Paramétrer les filtres en fonction du type de donnée

### **Atelier : mettre en place des filtres**

## **Relation entre les entités**

Créer des relations/associations entre les entités  
Lier les données à l'aide des IRI (International Resource Identifier)  
Afficher les informations d'une relation de façon détaillée  
Insérer, modifier, supprimer les valeurs d'une relation et les valider  
Créer des liens dédiés pour accéder aux sous ressources

### **Atelier : créer une relation entre deux entités et afficher les données**

## **Sécurité**

Identifier les différents type d'authentification  
Créer une entité utilisateurs et un contrôleur pour l'authentification  
Configurer l'authentification dans le fichier `security.yaml`  
Gérer les différents type d'erreurs d'authentification  
Restreindre l'accès à l'API depuis les routes, les contrôleurs, les classes

### **Atelier : créer une authentification pour les utilisateurs**

### **Atelier : limiter l'accès en lecture et écriture aux utilisateurs non connecté**

## **Télécharger des fichiers**

Installer le composant `VichUploader`  
Manipuler les types fichier : `POST` et `multipart/form-data`

Créer un decoder et denormalizer personnalisé pour le multipart/form-data  
Documenter l'entité avec openapi

**Atelier : télécharger et récupérer des images**