Doctrine

Version 1.2.0 Niveau requis : 4/7



Mise en œuvre de Doctrine 2.6 (O.R.M. PHP)

www.pequignat.eu

Sommaire

1.1. OBJET.	I. F	PREAMBULE	3
I.II. PRE-REQUIS 3 I.III. VERSIONS DU DOCUMENT 3 I.IV. DOCUMENTS DE REFERENCE 3 II. QU'EST CE QU'UN ORM ? 4 II.I. DESCRIPTION. 4 II.I. DESCRIPTION. 4 II.I. DESCRIPTION. 4 II.I. DESCRIPTION. 4 II.I. CREATION DU PROIET « COURS » 4 III.I. CREATION DU VIRUTAL HOST « COURS » 77 III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.II. INSTALLATION DE LA LIBRAINE DOCTRINE 5 III.II. NASTALATION DE LA LIBRAINE DOCTRINE 5 III.II.2 Récupération de Doctrine 13 III.V POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 13 III.V.1 Configuration d'accès à la base de données 15 III.V.1 Configuration d'accès à la base de données 15 III.V.1 Configuration d'accès à la base de données 15 III.V.1 Configuration d'accès à la base de données 15 III.V.1 Configuration d'accès à la base 15 </th <th>1.1.</th> <th>Овјет</th> <th>3</th>	1.1.	Овјет	3
I.III. VERSIONS DU DOCUMENT.	1.11.	Pre-requis	3
I.IV. DOCUMENTS DE REFERENCE 33 II. QU'EST CE QU'UN ORM ? 44 II.I. DESCRIPTION 45 III.I. CREATION DU PROJET « COURS » 55 III.I.I. CREATION DU VIDITAL HOST « COURS » 75 III.I.I. INSTALLATION DE LA UBBRAIRE DOCTRINE 75 III.I.I. Aloritation de la prise en charge de Composer 76 III.I NOSTIONNEMENT DE L'ENVIRONNEMENT & DEVELOPMENT » 135 III.V. CONFIGURATION DE DOCTRINE 135 III.V. CONFIGURATION DE DOCTRINE 135 III.V. CONFIGURATION DE SUBORTINE	1.111.	. Versions du document	3
II. QU'EST CE QU'UN ORM ? 4 II.I. DESCRIPTION 4 II.I. DESCRIPTION 4 III.I. DESCRIPTION 4 III.I. DESCRIPTION 4 III.I. DESCRIPTION 4 III.I. CREATION DU PROJET « COURS » 5 III.I. CREATION DU VIRUTAL HOST « COURS » 77 III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.II. INSTALLATION DE LA LIBRAIRIE DOCTRINE 8 III.II. NISTALIZION DE LA LIBRAIRIE DOCTRINE 8 III.II. Alout de la prise en charge de Composer 62 III.II. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 13 III.V. CONFIGURATION DE DOCTRINE 13 III.V. CONFIGURATION DE COLTRINE 15 III.V.1 Configuration d'accès à la base de données 15 III.V.2 Création du bocterma cours bans La BASE 15 IV.1. CONFIGURATION DU SCHEMA DE LA BASE 15 IV.1. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.I. CREATION DU SCHEMA CONS DA	I.IV.	. DOCUMENTS DE REFERENCE	3
II.I. DESCRIPTION	II. C	QU'EST CE QU'UN ORM ?	4
II.II. EXEMPLE PERSONNE - COURS 44 III. INSTALLATION DE DOCTRINE 5 III.I. CREATION DU VRUTAL HOST « COURS » 77 III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.II. Najout de la prise en charge de Composer 88 III.II. Ajout de la prise en charge de Composer 88 III.II. POSTIONNEMENT DE L'ENVRONNEMENT « DEVELOPMENT » 13 III.V. COnfiguration d'accès à la base de données. 15 III.V. COnfiguration d'accès à la base de données. 15 III.V. Configuration d'accès à la base de données. 15 III.V. Configuration d'accès à la base de données. 15 III.V. Configuration DU SCHEMA DE LA BASE 15 IV.I. CREATION DU SCHEMA DE LA BASE 15 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES 22 IV.V. CREATION DU SCHEMA DE LA BASE 23	11.1.	Description	4
III. INSTALLATION DE DOCTRINE 5 III.II. CREATION DU PROJET « COURS » 5 III.II. CREATION DU VIRUTAL HOST « COURS » 7 III.III. CREATION DU VIRUTAL HOST « COURS » 7 III.III. CREATION DU VIRUTAL HOST « COURS » 7 III.III. CREATION DE LA LIBRAIRIE DOCTRINE 8 III.III. Algout de la prise en charge de Composer 8 III.III. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 10 III.V. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 11 III.V. CONFIGURATION DE DOCTRINE 15 III.V.1 Configuration d'accès à la base de données 15 III.V.2 Création du boatstrap.php. 16 IV. CREATION DU SCHEMA DE LA BASE 15 IV.1. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES 22 IV.V. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES 22 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES 22 IV.V. CREATION DU SCHEMA DELABASE 22	11.11.	. Exemple Personne - Cours	4
III.I. CREATION DU PROJET « COURS »	III. I	INSTALLATION DE DOCTRINE	5
III.II. CREATION DU VIRUTAL HOST « COURS » 77 III.III. INSTALLATION DE LA LIBRAIRIE DOCTRINE 88 III.III.1 Ajout de la prise en charge de Composer 88 III.III.2 Récupération de Doctrine 106 III.V. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 13 III.V. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT » 13 III.V. CONFIGURATION DE DOCTRINE 15 III.V.1 Configuration d'accès à la base de données 15 III.V.2 Création du bootstrap.php. 16 IV. CREATION DU SCHEMA DE LA BASE 19 IV.1 CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.1 CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 21 IV.1 CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 22 IV.1 CREATION DE ENTITY SANS METTRE LES RELATIONS 22 IV.1. CREATION DE SENTTY SANS METTRE LES RELATIONS 23 IV.V.1 Relation ManyToOne 27 IV.V.1 Relation ManyToOne 27 IV.V.1 Relation ManyToOne 22 IV.V.2<	111.1.	. Creation du Projet « Cours »	5
III.III. INSTALLATION DE LA LIBRAIRIE DOCTRINE	.	I. CREATION DU VIRUTAL HOST « COURS »	7
III.III.1 Ajout de la prise en charge de Composer	.	II. INSTALLATION DE LA LIBRAIRIE DOCTRINE	8
III.III.2 Récupération de Doctrine	L	III.III.1 Ajout de la prise en charge de Composer	8
III.IV. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT »	L	III.III.2 Récupération de Doctrine	
III.V. CONFIGURATION DE DOCTRINE	111.1\	V. POSITIONNEMENT DE L'ENVIRONNEMENT « DEVELOPMENT »	13
III.V.1 Configuration d'accès à la base de données	III.V	/. CONFIGURATION DE DOCTRINE	15
III.V.2 Création du bootstrap.php. 16 IV. CREATION DU SCHEMA DE LA BASE 19 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU FICHIER CLI-CONFIG.PHP. 21 IV.III. LANCEMENT DANS LA CONSOLE MSDOS 22 IV.VI. CREATION DES ENTITY SANS METTRE LES RELATIONS 23 IV.V. CREATION SE ENTITY SANS METTRE LES RELATIONS 23 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES 26 IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI.1 Relation ManyTOOne 27 IV.VI.2 Relation ManyToMany 22 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.I. OBJECTIF 36 V.I. MISE EN QUURE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES DONNEES 44	L	III.V.1 Configuration d'accès à la base de données	15
IV. CREATION DU SCHEMA DE LA BASE 19 IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 21 IV.II. CREATION DU FICHIER CLI-CONFIG.PHP. 21 IV.III. LANCEMENT DANS LA CONSOLE MSDOS. 22 IV.V. CREATION DES ENTITY SANS METTRE LES RELATIONS. 23 IV.V. CREATION MES ENTITY SANS METTRE LES RELATIONS. 23 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation ManyToOne 27 IV.VI.2 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.I. OBJECTIF 36 V.I. MISE EN PLACE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES NOTRES 44 VI. SOURCES D'INFORMATIONS 44	L	III.V.2 Création du bootstrap.php	16
IV.I. CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL 19 IV.II. CREATION DU FICHIER CLI-CONFIG.PHP. 21 IV.III. LANCEMENT DANS LA CONSOLE MSDOS 22 IV.IV. CREATION DES ENTITY SANS METTRE LES RELATIONS 23 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES 26 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES 27 IV.V. VI.I. Relation ManyToOne 27 IV.VI.1 Relation inverse OneToMany 29 IV.VI.2 Relation ManyToOne 23 V. MISE EN CEUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.I. OBJECTIF 36 V.II. MISE A JOUR DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY. 39 V.V. MISE EN PLACE DE LA COUCHE SERVICE. 40 V.V. AFFICHAGE DES DONNEES 44 VI. SOURCES D'INFORMATIONS. 46	IV. C	CREATION DU SCHEMA DE LA BASE	19
IV.II. CREATION DU FICHIER CLI-CONFIG.PHP. .21 IV.III. LANCEMENT DANS LA CONSOLE MSDOS. .22 IV.IV. CREATION DES ENTITY SANS METTRE LES RELATIONS. .23 IV.V. CREATION SCE QUI S'EST PASSE EN BASE DE DONNEES. .26 IV.VI. MISE EN PLACE DES RELATIONS .27 IV.VI. MISE EN PLACE DES RELATIONS .27 IV.VI. Relation ManyToOne .27 IV.VI.2 Relation inverse OneToMany .25 IV.VI.3 Relation ManyToMany .25 IV.VI.3 Relation ManyToMany .32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES .36 V.I. Objectif	IV.I.	CREATION DU SCHEMA COURS DANS LA BASE DE DONNEES MYSQL	
IV.III. LANCEMENT DANS LA CONSOLE MSDOS. 22 IV.IV. CREATION DES ENTITY SANS METTRE LES RELATIONS. 23 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 27 IV.VI. Relation ManyToOne 27 IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation ManyToMany 29 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.I. OBJECTIF 36 V.I. OBJECTIF 36 V.I. Alout DES « REPOSITORY » 36 V.II. Alout DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY. 39 V.IV. MISE EN PLACE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES DONNEES 44 VI. SOURCES D'INFORMATIONS. 46 VII. FIN DU DOCUME	IV.II	I. CREATION DU FICHIER CLI-CONFIG.PHP	21
IV.IV. CREATION DES ENTITY SANS METTRE LES RELATIONS. 23 IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation inverse OneToMany 29 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. Objectif 36 V.I. Objectif 36 V.II. Ajout Des « REPOSITORY » 36 V.IV. Mise a jour Des Entity pour rajouter la declaration Des Repository. 39 V.IV. Mise En Place De La Couche Service 40 V.V. Affichage Des Donnees 44 VI. SOURCES D'INFORMATIONS 46	IV.II	II. LANCEMENT DANS LA CONSOLE MSDOS	22
IV.V. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES. 26 IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation inverse OneToMany 29 IV.VI.3 Relation ManyToMany. 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES. 36 V.I. Objectif 36 V.I. Objectif 36 V.I. MISE a Jour des Entity pour rajouter la declaration des Repository 39 V.IV. Mise en place de la couche Service 40 V.V. Africhage des donnees 44 VI. SOURCES D'INFORMATIONS 46	IV.IV	V. CREATION DES ENTITY SANS METTRE LES RELATIONS	23
IV.VI. MISE EN PLACE DES RELATIONS 27 IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation inverse OneToMany 29 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.II. AJOUT DES « REPOSITORY » 36 V.III. AJOUT DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY 39 V.IV. MISE EN PLACE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES DONNEES 44 VI. SOURCES D'INFORMATIONS 46 VII. FIN DU DOCUMENT 46	IV.V	/. VERIFIONS CE QUI S'EST PASSE EN BASE DE DONNEES	26
IV.VI.1 Relation ManyToOne 27 IV.VI.2 Relation inverse OneToMany 29 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES. 36 V.I. OBJECTIF 36 V.II. AJOUT DES « REPOSITORY ». 36 V.III. MISE A JOUR DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY. 39 V.IV. MISE EN PLACE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES DONNEES. 44 VI. SOURCES D'INFORMATIONS. 46 VII. FIN DU DOCUMENT 46	IV.V	/I. MISE EN PLACE DES RELATIONS	27
IV.VI.2 Relation inverse OneToMany 29 IV.VI.3 Relation ManyToMany 32 V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES 36 V.I. OBJECTIF 36 V.II. AJOUT DES « REPOSITORY » 36 V.III. MISE A JOUR DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY 39 V.IV. MISE EN PLACE DE LA COUCHE SERVICE 40 V.V. AFFICHAGE DES DONNEES 44 VI. SOURCES D'INFORMATIONS 46 VII. FIN DU DOCUMENT 46	1	IV.VI.1 Relation ManyToOne	27
IV.VI.3 Relation ManyToMany	1	IV.VI.2 Relation inverse OneToMany	29
V. MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES. 36 V.I. OBJECTIF 36 V.II. AJOUT DES « REPOSITORY ». 36 V.III. MISE A JOUR DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY. 39 V.IV. MISE EN PLACE DE LA COUCHE SERVICE. 40 V.V. AFFICHAGE DES DONNEES. 44 VI. SOURCES D'INFORMATIONS. 46 VII. FIN DU DOCUMENT 46	1	IV.VI.3 Relation ManyToMany	
V.I. Objectif 36 V.II. Ajout des « Repository »	V. M	MISE EN ŒUVRE DE L'ACCES A LA BASE DE DONNEES	36
V.II. Ajout des « Repository »	V.I.	Овјестіг	36
V.III. Mise a jour des Entity pour rajouter la declaration des Repository	V.II.	. Ajout des « Repository »	36
V.IV. Mise en place de la couche Service 40 V.V. Affichage des donnees 44 VI. SOURCES D'INFORMATIONS 46 VII. FIN DU DOCUMENT 46	V.III	I. MISE A JOUR DES ENTITY POUR RAJOUTER LA DECLARATION DES REPOSITORY	39
V.V. AFFICHAGE DES DONNEES	V.IV	/. Mise en place de la couche Service	40
VI. SOURCES D'INFORMATIONS	V.V.	AFFICHAGE DES DONNEES	44
VII. FIN DU DOCUMENT	VI. S	SOURCES D'INFORMATIONS	46
	VII. F	FIN DU DOCUMENT	46

I. Préambule

I.I. Objet

L'objet de ce document est de présenter l'approche de mise en œuvre de l'Object Relational Mapper (O.R.M.) Doctrine 2.6 en PHP.

I.II. **Pré-requis**

Avoir un environnement de développement installé avec PHP7, base de données MySQL 5. Nous allons partir du principe que vous avez suivi les documents suivants :

- 1. Installation Wamp_1.0.2
- 2. Installation Zend Studio_1.0.1
- 3. Debugger en PHP_1.0.1

Dans cette formation, nous utiliserons Zend Studio 13.6 Avoir téléchargé HeidiSQL : sur <u>https://www.heidisql.com/</u>

I.III. Versions du document

Version	Date	Auteur	Description
1.0.0	24/03/2018	Péquignat.eu	Version initiale du document
1.0.1	27/03/2018	Péquignat.eu	Mise en place de l'accès à la base de données
1.2.0	05/03/2022	Péquignat.eu	Retrait de l'autoentreprise

I.IV. Documents de référence

#	Document	Version	Auteur(s)
[R1]	Installation Wamp	1.0.2	Péquignat.eu
[R2]	Installation Zend Studio	1.0.1	Péquignat.eu
[R3]	Debugger en PHP	1.0.1	Péquignat.eu
[R4]	Virtual Host Wamp	1.0.1	Péquignat.eu

II. Qu'est ce qu'un ORM ?

II.I. Description

L'Object Relational Mapping est une petite architecture sous forme d'API permettant de faire le lien entre le modèle Objet représentant ses entités de travail et la base de données. Ce moteur permet de travailler sous forme d'une représentation Objet avec ses différentes relations entre Objets :

- 1..1 : One to One
- 1..N : Many to One
- N..1 : One to Many
- N..N : Many To Many

Avec les objets et ses relations, l'ORM fait l'interface avec la base de données (Ex : MySQL) pour assurer la persistance.

II.II. Exemple Personne - Cours

Nous allons présenter un exemple de mise en place d'un modèle objet pour ensuite avoir sa représentation en base de données.



Figure 1 - Personne - Cours

Nous avons ici deux entités : une Entité Personne et une Entité Cours.

Entre ces entités, nous avons plusieurs relations :

- ⇒ Cours qui est rédigé par une et une seule Personne : **OneToMany**
- ⇒ Relation Personne qui rédige zéro ou plusieurs Cours : ManyToOne
- ➡ Un Cours qui peut être suivi par plusieurs Personnes et une Personne qui peut suivre plusieurs Cours : ManyToMany

Déquiquet qu	Lot's build our futural	Version 1.0
requignal.eu		Le 05/03/2022

La mise en œuvre en Modèle de table est la suivante :



Figure 2 - Modèle de base de données

Nous avons ici trois tables, car en effet nous avons le relation ManyToMany qui est représentée par une table de lien PERSON_COURSE.

La relation ManyToOne d'une Personne vers le Cours et donc le OneToMany du Cours vers la Personne est représentée par une clé étrangère AUTHOR_FK dans la table COURSE qui fait référence à PERSON_ID de la table PERSON.

III. Installation de Doctrine

III.I. Création du Projet « Cours »

Créer un projet « PHP Local Project » Cours comme suit :

🕘 New Local PHP Project — 🗆 🗙							
Create a Local PHP Project							
		Settings			\rangle		
Project Name:	Cours						
Location:	C:\wamp64\www						
PHP Server:	Default PHP We	eb Server					
	Apache 2.4.27 - Lo	ocalhost		\sim A	Add Server		
Content:	Basic	O Zend Framework					
Version:	7.1 (void return ty	pe, class constant visibilit	y,)		~		
?	< Back	Next >	Finish	Ca	ancel		

Figure 3 - Création Projet Cours

Le projet s'est créé :

Péquignat.eu	Let's b	uild our future!	Version 1.0 Le 05/03/2022			
		Project Explorer 🛙	I Remote Systems	-		
		✓ 🗳 Cours		E \$	▽	
		> 🛋 PHP Languag > 🛋 PHP Include P > 🖻 index.php	e Library [PHP 7.1] Path			
		Figure	4 - Projet Cours créé			

Cette répartition du fichier index.php à la racine du répertoire cours n'est pas optimale. En effet, nous allons créer des sous répertoire à Cours comme « vendor », et une mauvaise utilisation pourrait laisser des failles.

Aussi nous allons créer un sous répertoire de « Cours » nommé « web ».

Nous y déplaçons le fichier index.php dans ce répertoire « web ».

len New Folder			\times
Folder Create a new folder resource.			-
Enter or select the parent folder:			
Cours			
 Image: Image: Im			
Folder name: web			
Advanced >>			
?	Finish	Cancel	

Figure 5 - Création du répertoire web

Cliquez sur Finish

Faite ensuite un déplacement du fichier index.php dans ce nouveau répertoire.

Péquignat.eu	Let's build our future!	Version 1.0 Le 05/03/2022
	V 🐸 Cours	
	> 🛋 PHP Language Library [PHP 7.1]	
	> 🛋 PHP Include Path	
	✓ ఊ web	
	> 🖻 index.php	

Figure 6 - Déplacement index.php

III.II. Création du Virutal Host « Cours »

Allez dans http://localhost

Cliquez sur Ajouter un Virtual Host : Nom : Cours Chemin : C:/wamp64/www/Cours/web/

Attention, mettre volontairement le sous répertoire « web » comme point d'entrée.

WampServer				
			• Version 3.1.0 -	64bit french
Apache Virtual Hosts 😋	:/wamp64/bin/apache/apache2.4.27/	/conf/extra/httpd-vhosts.conf		
VirtualHost déjà définis	:			
ServerName : localhost -	Directory : c:/wamp64/www	rld	Formulaire de suppr	ession de VirtualHost
Windows hosts C:\Win	dows\system32\drivers\etc\hosts			
🥜 Nom du Virtu	al Host Pas de caractères diacritiques	(éçëñ) - Pas d'espace - Pas de tiret ba	5 (_) Requis	
Cours				
🥜 Chemin compl	et absolu du dossier VirtualHost - Ex	emples : C:/wamp/www/projet/ ou E:/v	www/site1/ <mark>Requis</mark>	
C:/wamp64/www/C	ours/web/			

Figure 7 - Ajout du Virtual Host Cours

Cliquez sur « Démarrer la création du Virtual Host »

Péquignat.eu	Let's build our future!	Version 1.0 Le 05/03/2022
WampServer	Ajouter un VirtualHost - Retour à l'accueil	Version 3.1.0 - 64bit french
	Les fichiers ont été modifiés, le virtual host Cours a	i élé créé
Messages de	e la console pour actualisation des DNS :	
Vous pouvez a Cependant, po Redémannage du menu Outil	ijouter un autre VirtualHost en validant "Ajouter un VirtualHost" our que ces nouveaux VirtualHost soient pris en compte par Apache, vo DNS s par Clic-Droit sur l'icône Wampmanager. <i>(Cecl ne peut, hélas, pas ét</i>	ous devez lancer l'item tre failt automatiquement)

Figure 8 - Création réussi du VH

Redémarrer les services.

Vous pouvez cliquer sur http://cours

Vous devriez avoir une page blanche.

🗋 cours	×	Christopha —		×
$\leftrightarrow \ \Rightarrow \ G$	i cours/		☆ 🕸	:

Figure 9 (Page Cours

III.III. Installation de la librairie DocTrine

III.III.1 Ajout de la prise en charge de Composer

Nous allons utiliser « Composer » fourni avec Zend Studio afin de récupérer dans son projet : Doctrine en version 2.6.

Pour cela, faîtes un clique droit sur la souris sur le projet Cours, puis aller dans « Configure » et enfin cliquer sur « Add Composer Support »

uignat.eu	Let's build our future!

눱 Proje	ct E	xplorer 🖾 📲 Remote Systems		<u>ا</u>	Nelcome 2	2			
00			🖻 🔄 🎽		비노				
✓ 2 C	nure.	New Go Into		>	Velcome	zend studio 13.6			
2 🖷					etting started with PHP Getting started with Zend S				
v Ç		Show In	Alt+Shift-	+W >					
т н	_	Show in Local Terminal				Create a new PHP project from local files in remo			
		Сору	Ctrl	+C	oho	repository			
		Paste	Ctrl	+V	Prop				
	^	Refactor	Der	ele >	22	Create APIs with Apigility			
		lesset		,	11	Learn more			
		Import			5				
		Defrech				Create a new Zend Framework 2 project			
		Close Project			Z	Learn more			
		Close Uprelated Projects							
		Zeed Teel			-				
	25	Zend Iool							
		Show in Remote Systems view							
	N	Search	Ctrl	+H					
		Run As		>					
		Debug As		>					
		Profile As		>					
		Restore from Local History							
		Encode Project							
		Team		>					
		Compare With		>	<u> </u>				
		Configure		>	o ^{er} Add	Application Deployment Support			
		Source		>	Conv	vert to Faceted Form			
	_	Properties	Alt+En	ter	Conf	igure and Detect Nested Projects			
					Add	Composer Support			
					Conv	ert to Ja Add Composer Support			
					Rem	ove JavaScript Support			

Figure 10 - Ajout de la prise en charge de Composer

Un nouveau fichier s'est créé : « composer.json » à la racine du projet « Cours ». Ce fichier sert à gérer le projet et les dépendances.

Péqu

Péquignat.eu	Le	t's build our future!	Versio Le 05/	n 1.0 03/2022	
DefaultWorkspace - PHP - Cours/composer ji File Edit Navigate Search Project Run V Project Explorer II → III → III → V Project Explorer III → III → III → V Cours → MPHP Language Library [PHP 7.1]	son - Zend Window F R R R R R R R R R R R R R R R R R R R	Studio - D\Zend\workspace\DefaultWorkspace telp		Configuration	- D X
> all PIP Include Path ✓ @ web ⇒ @ index.php		Inis section describes general information about this package. Name: cours/cours Description: Type: Keywords: Homenage: License: Minimum Stability: stable Authors Honor the glorious authors of this package. Add_ Edit_ Remove Test 0	~	Configure ackage:	_
		Support Provide support options to your end-users. Email: Issues: Ecoum: Overview Dependencies Autoload Configuration Dependency Graph composer json Overview Dependencies Autoload Configuration Dependency Graph composer json Image: Console The Party Service State Stat	3	Schema Reference Lissues: Report Issues Reckagist: Browse Packages	¥. *** 0

Figure 11 - composer.json

III.III.2 Récupération de Doctrine

Allez dans la vue « composer.json » et rajouter les informations suivantes :

On va renommer le projet en « pequignat.eu/cours »

Dans la partie « require » : c'est la que nous demandons la dernière version 2.6.* existante



Figure 12 - configuration de Doctrine

```
{
    "name" : "pequignat.eu/cours",
    "require": {
        "doctrine/orm": "2.6.*"
    }
}
```

Déquienct ou	Let's build our future!	Version 1.0
r equignat.eu		Le 05/03/2022

Faite un Crtl+S pour sauvegarder les modifications.

Afficher ensuite la « View » Console comme suit :

DefaultWorkspace - PHP - Cours/composer.json - Zend Studio - D:\Zend\workspaces\DefaultWorkspace

File Edit Navigate Search Project Run	Window H	elp					
r → 📰 🐚 : 📮 : 🔌 🕨 💷 🖷 🕅 🔍 . 🛪 . C Project Explorer 🛱 🔏 Remote Systems	New Window Editor Appearance Show View		8	Image: Image			
x 🛱 Cours				% Code Coverage			
PHP Language Library (PHP 7.1)	Perspec	tive >		Console	Alt+Shift+Q, C		
> A PHP Include Path	Navigat	tion >	> HVC Outline Outline		Alt+Shift+Q, O	ige.	
 ✓ [™] web index.php composer.json HelloWorld 	Preferer	Description Type: Keywords Homepar License: Minimum Authors		 PHP Explorer PHP Functions PHP Project Outline PHP Servers PHPUnit Problems Project Explorer Remote Systems Tasks Type Hierarchy 	Alt+Shift+Q, X		
		Honor th	e	Other	Alt+Shift+Q, Q		
	Figure 13 - A	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	onso	ble			

Lancer un « Update Dependencies » sur la dépendance doctrine/orm:2.6.*

Welcome			- D
Dependencies			🐁 🌢 🍇 🛷 문
✓ Require The dependencies for your presented on the second s	package.		Packagist Search Search for packages and add the selected packages to the opened section on the left.
u doctrine/orm : 2.6.*	Updating Composer dependencies Updating Composer dependencies Always run in background Run in Background Run in Background Run in Background	Edit Remove	in the background where possible
Require (Development) Overview Dependencies Aut Problems Tasks © Cor Composer (Cours) Loading composer repos Package operations: 11 Installing symfony	oload Configuration Dependency Graph composer,son nsole ☎ m PHP Servers	_ Total: 1 utput	Add Dependencies

Figure 14 - Update Dependencies

Cela a dût créer le répertoire « vendor » contenant :

Péquignat.eu	Let's build our future!	Version 1.0 Le 05/03/2022		
	 Cours PHP Language Library [PHP 7.1] PHP Include Path verdor vendor verdor verd			

Maintenant rajouter dans le composer.json :



Faîtes un « Update dependencies »

Et créer les répertoires à la racine de Cours : config et src

Péquignat.eu	Let's build our future!	Version 1.0 Le 05/03/2022
	 Project Explorer Project Explorer Project Explorer Project Explorer Project Explorer Cours PHP Language Library [PHP 7.1] PHP Include Path Config Src Src Vendor Vendor Vendor Vendor Composer.json Composer.lock Composer.lock Composer.phar 	
	Figure 16 - Répertoire src et config	



Dans le répertoire Cours/web créé un fichier nommé « .htaccess ».

New File			×
File Create a new file resource.			
Enter or select the parent folder:			
Cours/web			
 Advanced >> Advanced >> Cours Settings 			
٢	Finish	Can	cel

Figure 17 - création du .htaccess

Déguige at au		Version 1.0
Pequignai.eu	Let's build our future!	Le 05/03/2022

Activer la vue « Navigator » dans « Windows » → « Show View »

Show View		\times
navi		R
✓ ➢ General Image: Second		
OK	Cancel	I

Figure 18 - Vue Navigator

Cliquer sur OK

Le fichier « .htaccess » est maintenant visible.



Allez dans le fichier « .htaccess » est rajouter la ligne :

SetEnv APPLICATION_ENV development

Déquianat au	Lotto build our future l	Version 1.0
requignat.eu		Le 05/03/2022

On se servira de « APPLICATION_ENV » pour déterminer sur quel environnement on se situe.

Valeurs préconisées : « development » ou « production ».

Rajouter maintenant dans l'index.php de web le contenu suivant :

php</td
defined ("APPLICATION_ENV") define("APPLICATION_ENV",
<pre>getenv('APPLICATION_ENV') ?? 'production');</pre>
echo APPLICATION ENV;

Et afficher la page : <u>http://cours</u>

🗋 cours	×	Christophe	_		×
\leftrightarrow \rightarrow C (i) cours/				☆ 🏘	:
development					

Figure 20 - affichage de l'environnement

Attention : chez un Hébergeur tel qu'OVH, le passage de variable d'environnement est désactivé sur l'hébergement mutualisé.

Il convient dans l'utilisation de définir une valeur par défaut étant la « production » dans le code PHP.

III.V. Configuration de Doctrine

III.V.1 **Configuration d'accès à la base de données**

Créons un fichier config/Database.php contenant les accès à la base de données MySQL.

<?php class Database { /**

```
Péquignat.eu
```

```
* Renvoie les paramètres de connexion
* @return array
*/
public static function getConnectionParams() : array
{
    return array(
        'dbname' => 'cours', // Le nom du schema de la base
        'user' => 'root', // L'utilisateur de connexion
        'password' => '', // Le mot de passe
        'host' => 'localhost', // Le serveur ici localhost
        'driver' => 'pdo_mysql' // Le driver de connexion PDO
    );
}
```

Créons aussi un fichier « .htaccess » présent dans ce même répertoire pour le protégé : Avec le contenu :





III.V.2 Création du bootstrap.php

La classe Bootstrap.php est un fichier qui initialise toute l'application. Nous allons la créer dans le répertoire « src ».

Nous y allons y déclarer l'initialisation de la base de données. C'est là aussi que l'on peut initialiser la session PHP pour les sites ayant une session ou encore les Log avec Log4PHP...



```
Version 1.0
               Let's build our future!
  Péquignat.eu
                                                                  Le 05/03/2022
use Doctrine\ORM\EntityManager;
class Bootstrap{
    /**
     *
     * @var bool
    */
    private static $isDevMode = false;
    /**
     *
     * @var EntityManager
     */
   private static $entityManager = null;
    private static function initDevMode() : void {
        self::$isDevMode = in array( APPLICATION ENV , array('development'));
    }
   private static function initDatabase() : void{
        // Create a simple "default" Doctrine ORM configuration for Annotations
        $config = Setup::createAnnotationMetadataConfiguration(array( DIR ),
self::$isDevMode);
        // or if you prefer XML
        //$config =
Setup::createXMLMetadataConfiguration(array( DIR ."/config"), $isDevMode);
        // database configuration parameters
        $conn =
\Doctrine\DBAL\DriverManager::getConnection(Database::getConnectionParams(),
$config);
        // obtaining the entity manager
        self::$entityManager = EntityManager::create($conn, $config);
    }
```

```
Péquignat.eu
```

```
public static function init() : void{
    self::initDevMode();
    self::initDatabase();
}
/**
 * Donne access à l'EntityManager d'accès à la base
 * @return EntityManager
 */
public static function getEntityManager() : EntityManager{
    return self::$entityManager;
}
```

La méthode Bootstrap::init() initialise tout ce qui faut pour le démarrage.

Voyons maintenant le fichier bootstrap.php se situant à la racine du projet



Revenons au fichier index.php présent dans web...

Mettre le contenu suivant :

```
<?php
defined ("APPLICATION_ENV") || define("APPLICATION_ENV",
getenv('APPLICATION_ENV') ?? 'production');
echo APPLICATION_ENV . "<br/>\r\n";
echo 'Initialisation de la connexion à la base de données';
require_once('..\bootstrap.php');
```

| 🗋 cours | × |
|-------------------------------------|--------------------------------------|
| $\leftrightarrow \ \Rightarrow \ G$ | (i) cours/ |
| development
Initialisation | de la connexion à la base de données |

Figure 22 - Affichage de la connexion

IV.Création du schéma de la base

IV.I. Création du Schéma cours dans la base de données MySQL

| Connectez-vous sur le lien : h | nttp://localhost/phpmyadmin/ |
|--------------------------------|------------------------------|
|--------------------------------|------------------------------|

| // phpMyAdmin ★ | Christophe | _ | | × |
|---------------------------------|------------|----|-----|---|
| ← → C (i) localhost/phpmyadmin/ | | 07 | ☆ 🕸 | : |
| phpMyAdmin | | | | |
| Bienvenue dans phpMyAdmin | | | | |
| Langue - Language | | | | |
| Français - French | | | | |
| Connexion 😡 | | | | |
| Utilisateur : root | | | | |
| Mot de passe : | | | | |
| Choix du serveur : MySQL 🔻 | | | | |
| | | | | |
| Exécuter | | | | |
| | | | | |

Figure 23 - Connexion PHPMyAdmin

Cliquez sur Exécuter

| Déquianat au | Lot's build our futural | Version 1.0 |
|--------------|-------------------------|---------------|
| requignat.eu | | Le 05/03/2022 |

| Iocalhost / MySQL php 🛛 🗙 | | Christophe — 🗆 🗙 |
|--|---|--|
| ← → C () localhost/phpmyad | dmin/index.php | ☆ 🕸 🗄 |
| phpMyAdmin | Serveur: MySQL 3306 Bases de données Comptes utilisateurs Fronter Fronter | mporter 🏕 Paramètres 📗 Réplication 🔻 Plus |
| Récentes Préférées | Paramètres généraux Modifier le mot de passe Interclassement pour la connexion au serveur Utt8mb4_unicode_ci | Serveur de base de données
• Serveur : MySQL (127.0.0.1 via TCP/IP)
• Type de serveur : MySQL
• Version du serveur : 5.7.19 - MySQL Community Server
(GPL) |
| ● @ mysql
● @ performance_schema
● @ sys | Paramètres d'affichage | Version au protocole : 1 Ulitateur: root@loc.ub Jeu de caractères du serveur : UTF-8 Unicode (utf8) Serveur Web |
| | ④ Thème : pmahomme ▼ • Taille du texte: 82% ▼ | Apache/2.4.27 (Win64) PHP/7.1.9 Version du client de base de données : libmysql - mysqlnd
5.0.12-dev - 20150407 - \$ld:
b396954eeb2d1d9ed7902b8bae237b287f21ad9e \$ Extension PHP : mysqli @ curl @ mbstring @ Version de PHP : 7.1.9 |
| | Console de requêtes SQL | phpMyAdmin Version : 4.7.4 Documentation Site officiel |

Figure 24 - Accueil PHPMyAdmin

Cliquez sur Bases de données

Compléter le nom de la base de données : cours Encodage : utf8_general_ci



Figure 25 - Création du schéma de données

Cliquez sur « Créer » On est redirigé vers :

| Péquianat eu | et's build our future! | Version 1.0 |
|---|-------------------------|---|
| | | Le 05/03/2022 |
| Péquignat.eu
↓ localhost / MySQL / cour >
← → C ① localhost/p
phpMyAdmii | Let's build our future! | Le 05/03/2022
@ndstephe - □ ×
☆ * :
rter ♪ Opérations ▼ Plus
Exécuter |
| | Console de requêtes SQL | |

Figure 26 - Schéma cours créé

IV.II. Création du fichier cli-config.php

Doctrine a un applicatif en ligne de commande qui permet en autre de générer le schéma de la base de données Relationnel.

Cette génération se base sur les classe Entity de votre projet en utilisant les métadata (Annotation dans les commentaires).

Créez le fichier « cli-config.php » à la racine de Cours

```
<?php
// cli-config.php
defined ("APPLICATION_ENV") || define("APPLICATION_ENV", 'development');
require_once "bootstrap.php";
return
\Doctrine\ORM\Tools\Console\ConsoleRunner::createHelperSet(Bootstrap::getEntityM
anager());
```

Nous allons initialiser le bootstrap.php et récupérer l'EntityManager pour le passer à la console.

| Déquianat au | Lot's build our futural | Version 1.0 |
|--------------|-------------------------|---------------|
| requignatieu | | Le 05/03/2022 |

IV.III. Lancement dans la console MSDOS

Ouvrir une Commande « CMD » dans la barre de recherche de Windows.

Tapez :

cd /D c:\wamp64\www\Cours

Vous devriez voir ceci :

| 🔤 Invite de commandes | - | × |
|-----------------------|---|-----|
| | | ^ |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | J |
| | | · · |

Figure 27 - Commande MSDOS

Dans la console tapez maintenant :

set PATH=C:\wamp64\bin\php\php7.1.9;%PATH%

vendor\bin\doctrine.bat orm:schema-tool:create



Figure 28 - Lancement commande création schéma

Le résultat de la commande si tout est bien initialisé :



Figure 29 - Fin du processus de création des tables

Nous verrons après la création des Entity Person et Course avec leur méta données.

Une fois mise en place, deux méthodes sont possibles pour mettre à jour le modèle de la base :

vendor\bin\doctrine.bat orm:schema-tool:drop --force vendor\bin\doctrine.bat orm:schema-tool:create

ou

vendor\bin\doctrine.bat orm:schema-tool:update --force

IV.IV. Création des Entity sans mettre les relations

Créer dans le répertoire « src », un sous répertoire « Entity » et y créer les deux fichiers PHP :

• Person.php

| Déquiquent qu | Lotte build our future l | Version 1.0 |
|---------------|--------------------------|---------------|
| Pequignat.eu | | Le 05/03/2022 |

• Course.php

Voici le contenu de ces Entity :

Pour Person.php

```
<?php
namespace Entity;
/**
* @Entity
 * @Table(name="PERSON")
 * @author Christophe PEQUIGNAT
 */
class Person
{
    /**
    * @Id
     * @Column(type="integer", name="PERSON_ID")
     * @GeneratedValue
    * @var int
    */
    private $id;
    /**
    * @Column(type="string", length=255, name="NAME")
    * @var string
    */
    private $name;
    /** @Column(type="string", length=255, name="FIRST NAME")
    * @var string
     */
    private $firstname;
```

| Péquignat.eu | Let's build our future! | Version 1.0
Le 05/03/2022 |
|-------------------------|--|------------------------------|
| /** @Colum
* @var st | n(type="string", length=255, name="JOB")
ring | |
| <pre></pre> | ob; | |

Pour Course.php

```
<?php
namespace Entity;
/**
 * @Entity
 * @Table(name="COURSE")
* @author Christophe PEQUIGNAT
 *
*/
class Course
{
    /**
     * @Id
     * @Column(type="integer", name="COURSE")
     * @GeneratedValue
     * @var int
     */
    private $id;
    /**
     * @Column(type="string", length=1024, name="TITLE")
     * @var string
     */
    private $title;
```

}

Relançons la commande de mise à jour de la base de données :

vendor\bin\doctrine.bat orm:schema-tool:create



Figure 30 - Création du schéma sans relation OK

IV.V. Vérifions ce qui s'est passé en base de données

Cliquez sur le lien : <u>http://localhost/phpmyadmin/db_structure.php?server=1&db=cours</u>

| Péquignat.eu Let's build our future! | Lot's build our futural | Version 1.0 |
|--------------------------------------|-------------------------|---------------|
| | | Le 05/03/2022 |
| | | |

| Alago Alago | indelopino — | | × |
|--|--------------|---------|---------|
| ← → C ① localhost/phpmyadmin/db_structure.php?server=1&db=cours | | ☆ * | \$F : |
| phpMuAdmin - Inf Serveur: MySQL 3306 » 🗈 Base de données: cours | | | ☆ ⊼ |
| Image: Serveur courant : MySQL Récentes Préférées Préférées | ▼ Plus | | |
| Ignes 🕢 Type Interclassement Ta | ille Perte | | |
| cours Person Person Person Rechercher Schwerz Vider Supprimer einnobb utgemeerde ins | ,0 ki0 - | | |
| Currie 2 tables Somme • MyISAM utf8 general_ci 32, | ,0 kio 0 o | | |
| Colonnes Golonnes | | | |
| | E | xécuter | |
| montadon_concerto montadon_concert | | | |
| Eigure 21 Vérification de la base source cans relation | | | |

IV.VI. Mise en place des relations

IV.VI.1 Relation ManyToOne

La première relation à positionner est celle du ManyToOne

```
<?php
namespace Entity;
/**
* @Entity
* @Table(name="COURSE")
* @author Christophe PEQUIGNAT
*
*/
class Course
{
/**
* @Id
```

| Péquianat eu | Let's build our future | Version 1.0 |
|--------------------|--|---------------|
| i equignati.eu | | Le 05/03/2022 |
| | | |
| * @Column | (type="integer", name="COURSE") | |
| * @Genera | tedValue | |
| * @var in | t | |
| * / | | |
| private \$i | d; | |
| | | |
| /** | | |
| * @Column | (type="string", length=1024, name="TITLE") | |
| * @var st | ring | |
| * / | | |
| private \$t | itle; | |
| | | |
| | | |
| <mark>/ * *</mark> | | |
| * @ManyTo | One(targetEntity="Person") | |
| * @JoinCo | | N ID") |
| */ | | |
| private \$a | uthor; | |
| | | |
| 1 | | |
| J | | |
| | | |

Nous allons réinitialiser la base de données :



Figure 32 - Effacement de la base de données

Et nous relançons la création du schéma :

| Invite de commandes | - | × |
|--|---|---|
| c:\wamp64\www\Cours>vendor\bin\doctrine.bat orm:schema-tool:create | | |
| ! [CAUTION] This operation should not be executed in a production environment! | | |
| Creating database schema | | |
| [OK] Database schema created successfully! | | |
| c:\wamp64\www\Cours> | | |

Figure 33 - Création de la base de données avec relation ManyToOne

Voyons dans l'outil HeidiSQL ce qui s'est passé.



Figure 34 - Heidi SQL - Visualisation de la FK

IV.VI.2 Relation inverse OneToMany

Nous allons rajouter dans l'Entity Person, la liste des cours dont il est author.

Rajoutons dans la class Person



```
* @author Christophe PEQUIGNAT
*/
class Person
{
   /**
    * @Id
    * @Column(type="integer", name="PERSON ID")
    * @GeneratedValue
    * @var int
    */
   private $id;
   /**
    * @Column(type="string", length=255, name="NAME")
    * @var string
    */
   private $name;
   /** @Column(type="string", length=255, name="FIRST NAME")
    * @var string
    */
   private $firstname;
   /** @Column(type="string", length=255, name="JOB")
    * @var string
    */
   private $job;
   /**
  * Une Person rédige plusieurs Course.
   * @OneToMany(targetEntity="Course", mappedBy="author")
 */
private $myCourses;
}
```

| Péquignat.eu | Lat's build our futural | | Version 1.0 |
|--------------|-------------------------|---------------|-------------|
| | Let's build our future! | Le 05/03/2022 | |

Nous modifions aussi le cours pour indiquer la double relation :

```
<?php
namespace Entity;
/**
 * @Entity
 * @Table(name="COURSE")
 * @author Christophe PEQUIGNAT
 *
 */
class Course
{
    /**
    * @Id
     * @Column(type="integer", name="COURSE")
     * @GeneratedValue
     * @var int
    */
    private $id;
    /**
     * @Column(type="string", length=1024, name="TITLE")
     * @var string
     */
    private $title;
    /**
    * @ManyToOne(targetEntity="Person", inversedBy="myCourses")
     * @JoinColumn(name="AUTHOR FK", referencedColumnName="PERSON ID")
     */
    private $author;
```

| Péquignat.eu | Let's build our future! | Version 1.0 | |
|--------------|-------------------------|--------------|---|
| 1 5 | | Le 05/03/202 | 2 |

Rafraichissons la base de données.

| Invite de commandes — | | \times |
|--|------|----------|
| wamp64\www\Cours>vendor\bin\doctrine.bat orm:schema-tool:dropforce | | |
| opping database schema | | |
| <pre>K] Database schema dropped successfully!</pre> | | |
| wamp64\www\Cours>vendor\bin\doctrine.bat orm:schema-tool:create | | |
| [CAUTION] This operation should not be executed in a production environment! | | |
| eating database schema | | |
| <pre>WK] Database schema created successfully!</pre> | | |
| wamp64\www\Cours> | | |
| Figure 35 - Recréation de la base de données |
 | |

IV.VI.3 Relation ManyToMany

Nous avons des Cours que suivent des Personnes. Cette relation est une relation ManyToMany. En effet, chaque cours peut être suivi par plusieurs personnes, et Une Personne peut suivre plusieurs Cours.

Modifions la Class Person.php



```
*/
   private $id;
   /**
    * @Column(type="string", length=255, name="NAME")
    * @var string
    */
   private $name;
   /** @Column(type="string", length=255, name="FIRST NAME")
    * @var string
    */
   private $firstname;
   /** @Column(type="string", length=255, name="JOB", nullable=true)
    * @var string
    */
   private $job;
   /**
    * Une Person rédige plusieurs Course.
    * @OneToMany(targetEntity="Entity\Course", mappedBy="author")
    */
   private $myCourses;
   / * *
 * Plusieurs Persons suivent plusieurs cours.
   * @ManyToMany(targetEntity="Entity\Course", inversedBy="students")
* @JoinTable(name="PERSON_COURSE",
    * joinColumns={@JoinColumn(name="PERSON ID",
referencedColumnName="PERSON ID")},
  * inverseJoinColumns={@JoinColumn(name="COURSE ID",
referencedColumnName="COURSE ID")}
* )
*/
private $myTrainingCourses;
```

| Péquignat.eu | Let's build our future! | Version 1.0
Le 05/03/2022 |
|-----------------------------|--|------------------------------|
| | 1 | |
| | | |
| public fun | ctionconstruct() { | |
| \$this- | >myTrainingCourses = new | |
| <mark>\Doctrine\Comm</mark> | <pre>on\Collections\ArrayCollection();</pre> | |
| } | | |
| | | |
| } | | |

Et la classe Course.php

```
<?php
namespace Entity;
/**
 * @Entity
 * @Table(name="COURSE")
* @author Christophe PEQUIGNAT
 *
*/
class Course
{
    /**
    * @Id
    * @Column(type="integer", name="COURSE ID")
    * @GeneratedValue
    * @var int
    */
    private $id;
    /**
    * @Column(type="string", length=1024, name="TITLE")
    * @var string
    */
    private $title;
```

| Péquignat.eu | Let's build our future! | Version 1.0 |
|--------------------|---|---------------------------|
| | | Le 05/03/2022 |
| | | |
| | | |
| / * * | | |
| * @ManyTo | One(targetEntity=" <mark>Entity\</mark> Person", inversedBy="myCo | urses") |
| * @JoinCo | <pre>lumn(name="AUTHOR_FK", referencedColumnName="PERSO</pre> | N_ID", |
| nullable=false |) | |
| * / | | |
| private \$a | uthor; | |
| | | |
| <mark>/**</mark> | | |
| * Plusieu | rs cours ont plusieurs étudiants. | |
| * @ManyTo | Many(targetEntity="Entity\Person", mappedBy="myTra | <pre>iningCourses")</pre> |
| * / | | |
| private \$s | tudents; | |
| | | |
| public fun | ction construct() { | |
| \$this- | >students = new \Doctrine\Common\Collections\Array | Collection(); |
| } | | |
| | | |
| | | |
| } | | |

On relance avec l'option –dump-sql pour la partie création afin d'avoir la requête générée :



Figure 36 - Installation ManyToMany DB

On récupère le fichier se trouvant dans c:\wamp64\www\install.sql

The following SQL statements will be executed:

CREATE TABLE COURSE (COURSE_ID INT AUTO_INCREMENT NOT NULL, TITLE VARCHAR(1024) NOT NULL, AUTHOR_FK INT NOT NULL, INDEX IDX_E666A83F9DDDAA4A

| Péquignat.eu | Let' |
|--------------|------|
| 1 5 | |

(AUTHOR_FK), PRIMARY KEY(COURSE_ID)) DEFAULT CHARACTER SET utf8 COLLATE utf8 unicode ci ENGINE = InnoDB;

CREATE TABLE PERSON (PERSON_ID INT AUTO_INCREMENT NOT NULL, NAME VARCHAR(255) NOT NULL, FIRST_NAME VARCHAR(255) NOT NULL, JOB VARCHAR(255) DEFAULT NULL, PRIMARY KEY(PERSON_ID)) DEFAULT CHARACTER SET utf8 COLLATE utf8 unicode ci ENGINE = InnoDB;

CREATE TABLE PERSON_COURSE (PERSON_ID INT NOT NULL, COURSE_ID INT NOT NULL, INDEX IDX_63CA64765DF4E348 (PERSON_ID), INDEX IDX_63CA64762593919D (COURSE_ID), PRIMARY KEY(PERSON_ID, COURSE_ID)) DEFAULT CHARACTER SET utf8 COLLATE utf8 unicode ci ENGINE = InnoDB;

ALTER TABLE COURSE ADD CONSTRAINT FK_E6666A83F9DDDAA4A FOREIGN KEY (AUTHOR FK) REFERENCES PERSON (PERSON ID);

ALTER TABLE PERSON_COURSE ADD CONSTRAINT FK_63CA64765DF4E348 FOREIGN KEY (PERSON_ID) REFERENCES PERSON (PERSON_ID);

ALTER TABLE PERSON_COURSE ADD CONSTRAINT FK_63CA64762593919D FOREIGN KEY (COURSE_ID) REFERENCES COURSE (COURSE_ID);

V. Mise en œuvre de l'accès à la base de données

V.I. Objectif

L'objectif de cette partie est d'enrichir le site en mettant en place deux DAO (ou Repository) permettant l'accès respectivement au Cours et Personne.

V.II. Ajout des « Repository »

Dans le répertoire Cours/src, rajouter un nouveau répertoire « Repository ».

Nous allons créer deux nouvelles classes : CourseRepository ainsi que PersonRepository

Cours
PHP Language Library [PHP 7.1]
PHP Include Path
Config
Src
Entity
P Course.php
P Person.php
P Repository
P CourseRepository.php
P PersonRepository.php
Figure 37 - Ajout des deux Repository

Les deux classes doivent étendre de « EntityRepository » de Doctrine\ORM\EntityRepository.

<u>CourseRepository :</u>

| Pequignat.eu Let's build our future! | Péquignat.eu |
|--------------------------------------|--------------|
|--------------------------------------|--------------|

```
<?php
namespace Repository;
use Doctrine\ORM\EntityRepository;
/**
 * @author Christophe PEQUIGNAT
 */
class CourseRepository extends EntityRepository
{
    public function getAllCourses()
    {
        return $this->getEntityManager()->createQuery('SELECT c FROM
\Entity\Course c')
        ->getResult();
    }
    public function addCourse(\Entity\Course $course) {
        $this->getEntityManager()->persist($course);
    }
    public function updateCourse(\Entity\Course $course) {
        $this->getEntityManager()->persist($course);
    }
    public function removeAll() : void{
        foreach($this->getAllCourses() as $course){
            $this->getEntityManager()->remove($course);
        }
    }
}
```

| Déquiquent au | Latis build our futural | Version 1.0 |
|---------------|-------------------------|---------------|
| Pequignal.eu | Let's build our future! | Le 05/03/2022 |

PersonRepository :

<?php

```
namespace Repository;
use Doctrine\ORM\EntityRepository;
class PersonRepository extends EntityRepository
{
   public function getAllPersons()
    {
        return $this->getEntityManager()->createQuery('SELECT p FROM
\Entity\Person p')
        ->getResult();
    }
    public function addPerson(\Entity\Person $person) : void{
        $this->getEntityManager()->persist($person);
    }
    public function removeAll() : void{
        foreach($this->getAllPersons() as $person){
            $this->getEntityManager()->remove($person);
        }
    }
}
```

Vous remarquerez que les requêtes d'accès à la base ne sont pas directement écrit en SQL.

SELECT p FROM \Entity\Person p

C'est volontaire afin de mettre une abstraction sur l'écriture pour permettre plus facilement la migration entre base de données. Aussi il est plus facile de changer de base entre MySQL, Posgres, Oracle, sqlite...

Les Repository à mon sens ne doivent pas gérer la transaction de la base de données. C'est au niveau au dessus que doivent être gérer ces transactions : dans la couche Service.

V.III. Mise à jour des Entity pour rajouter la déclaration des Repository

Afin de pouvoir utiliser ces nouveaux Repository, ils doivent être reliés avec leur Entity.

Voici en jaune ce qui faut rajouter pour la prise en compte pour l'Entity Course :

```
<?php
namespace Entity;
use Doctrine\ORM\Mapping\Entity as Entity;
/**
 * @Entity{repositoryClass="Repository\CourseRepository")
 * @Table(name="COURSE")
 *
 * @author Christophe PEQUIGNAT
 */
class Course
{
//...
}</pre>
```

Et pour l'Entity Person :



| Péquignat.eu | Let's build our future! | Version 1.0
Le 05/03/2022 |
|---------------------|-------------------------|------------------------------|
| */ | | |
| class Person | | |
| { | | |
| // | | |
| } | | |

Comme vous pouvez le voir, on utilise maintenant l'annotation @Entity de Doctrine\ORM\Mapping\Entity. Cela permet de conserver le mapping à la base ainsi que d'avoir les options avec le Repository : repositoryClass="namespaces\ClassRepository").

Il convient de rajouter maintenant les getters et setters dans les Entity.

V.IV. Mise en place de la couche Service

Maintenant, il convient de rajouter les classes permettant de répondre aux exigences métiers de l'application et ceci dans un contexte transactionnel. C'est la couche Service !

Créer un répertoire Service dans src.



Rajouter une classe abstraite : <u>AbtractService.php</u>

Cette classe sert à définir les méthodes et attribut commun à tous les Services de l'application : à savoir contenir l'EntityManager.



| Péquignat.eu | Let's build our future! | Version 1.0
Le 05/03/2022 |
|-----------------|--|------------------------------|
| | | |
| abstract class | AbstractService | |
| /** | | |
| * @var En
*/ | tityManager | |
| protected | \$_em; | |
| protected | function setEntityManager(EntityManager <u>\$em</u>) : voi | d |
| \$this- | >_em = <u>\$em;</u> | |
| protected | <pre>function getEntityManager() : EntityManager{</pre> | |
| return | <pre>\$this->_em;</pre> | |
| } | | |
| , | | |

Et la classe CourseService.php

```
<?php
namespace Service;
use Doctrine\ORM\EntityManager;
use Entity\Person;
use Entity\Course;
/**
 * La classe de service qui gère de haut niveau l'aspect métier ainsi que les
transactions
 * Traitez les transactions au plus haut niveau possible pour mieux gérer les
plantages, oui cela arrive
 *
 * @author Christophe PEQUIGNAT</pre>
```

| Péar | ıian | at. | еu |
|------|------|-----|----|
| requ | uyn | uı. | cu |

```
*/
class CourseService extends AbstractService
{
   public function construct(EntityManager $em) {
        $this->setEntityManager($em);
    }
    /**
     * Initialise la base de données avec des valeurs ceci dans une transaction
unique
    */
   public function initExample() : void{
        $this->getEntityManager()->transactional(function(EntityManager $em){
            $author = new Person();
            $author->setFirstname("Christophe");
            $author->setName("Péquignat");
            $author->setJob("Software Architect");
            $course = new Course();
            $course->setAuthor($author);
            $course->setTitle('Mise en oeuvre de Doctrine 2.6');
            $author->addMyCourse($course);
            $student = new Person();
            $student->setFirstname("John");
            $student->setName("Doe");
            $student->setJob("Freelance");
            $em->getRepository('Entity\Person')->addPerson($author);
            $em->getRepository('Entity\Course')->addCourse($course);
            $em->getRepository('Entity\Person')->addPerson($student);
```

```
$course->addStudent($student);
            $student->addMyTrainingCourse($course);
            //En sortie de la méthode le relation ManyToMany est enregistrée en
base
       });
    }
    /**
     * Efface la base de données de toutes ses valeurs ceci dans une transaction
unique
    */
    public function dropExample() : void{
        $this->getEntityManager()->transactional(function(EntityManager $em){
            $em->getRepository('Entity\Course')->removeAll();
            $em->getRepository('Entity\Person')->removeAll();
       });
    }
    public function getAllPersons() {
        return $this->getEntityManager()->getRepository('Entity\Person')-
>getAllPersons();
    }
    public function getAllCourses() {
        return $this->getEntityManager()->getRepository('Entity\Course')-
>getAllCourses();
    }
}
```

Cet exemple, ce n'est pas conforme pour une utilisation réelle, en effet nous créons les données de Tests directement dans le service. Hors il conviendrait d'écrire des Tests Unitaires ou utiliser un mécanisme en dehors de l'application web pour initialiser les données.

| Dí | Latis build our futural | Version 1.0 | |
|--------------|-------------------------|-------------|---------------|
| Pequignai.eu | Let's build our future! | | Le 05/03/2022 |

V.V. *Affichage des données*

Maintenant que nous avons notre service d'appel, nous pouvons enfin réaliser l'affichage des données.

Voici le rendu lors que l'on lance la création de la base et dessuite une première fois la page <u>http://cours</u>

```
set PATH=C:\wamp64\bin\php\php7.1.9;%PATH%
```

vendor\bin\doctrine.bat orm:schema-tool:drop --force

vendor\bin\doctrine.bat orm:schema-tool:create



Figure 39 - Réinitialisation de la base de données

| Péquignat.eu | Let's build our future! | Version 1.0 |
|--------------|-------------------------|---------------|
| | | Le 05/03/2022 |
| | | |

| 🗋 cours | | × | Christoph | e — | | × | (|
|--|---|---|---|------------|-----|---|---|
| $\leftarrow \rightarrow G$ | i cours/ | | | | ☆ * | * | : |
| Environnem
Initialisation
Initialisation
Affichage de
==>Affichag
Person{id:1,
Person{id:2,
==>Affichag
Course{id:1 | ent : develop
de la connex
des datas
es datas
ge des Person
firstname:Ch
firstname:Joł
ge des Cours
title:Mise en | ment
tion à la base de c
nes :
ristophe,name:Pé
m,name:Doe,job:
ceuvre de Doctri | lonnées
quignat,job:Software A
Freelance}
ne | Architect} | | | |
| 2.6,author:Person {id:1,firstname:Christophe,name:Pequignat,job:Software
Architect}students:[Person{id:2,firstname:John,name:Doe,job:Freelance}]} | | | | | | | |
| Suppression des datas | | | | | | | |

Figure 40 - Rendu simple mais efficace

Le fichier<u>web\index.php</u>

```
<?php
defined ("APPLICATION_ENV") || define("APPLICATION_ENV",
getenv('APPLICATION_ENV') ?? 'production');
echo 'Environnement : '.APPLICATION_ENV . "<br/>\r\n";
echo 'Initialisation de la connexion à la base de données'. "<br/>\r\n";
require_once('..\bootstrap.php');
use \Service\CourseService;
$courseService = new CourseService(Bootstrap::getEntityManager());
echo 'Initialisation des datas'. "<br/>\r\n";
$courseService->initExample();
echo 'Affichage des datas'. "<br/>\r\n";
foreach($courseService->getAllPersons() as $person){
```

| Péquignat.eu | Let's build our future! | Version 1.0
Le 05/03/2022 | |
|---|-------------------------|------------------------------|--|
| echo \$pers | on."
\r\n"; | | |
| } | | | |
| echo "
\r\: | n"; | | |
| <pre>echo '==>Affichage des Cours :'."
\r\n";</pre> | | | |
| <pre>foreach(\$courseService->getAllCourses() as \$cours){</pre> | | | |
| <pre>echo \$cours."
>\r\n";</pre> | | | |
| } | | | |
| echo "
\r\n"; | | | |
| echo 'Suppression des datas'. "
\r\n"; | | | |
| <pre>\$courseService->dropExample();</pre> | | | |

VI.Sources d'Informations

| # | Source | Lien |
|------|------------------------|---|
| [S1] | Doctrine | http://docs.doctrine-project.org/projects/doctrine- |
| | | orm/en/latest/ |
| [S2] | Doctrine Association - | http://docs.doctrine-project.org/projects/doctrine- |
| | Mapping | orm/en/latest/reference/association-mapping.html |

VII. Fin du document