Symfony Partie 2
Relations entre entités

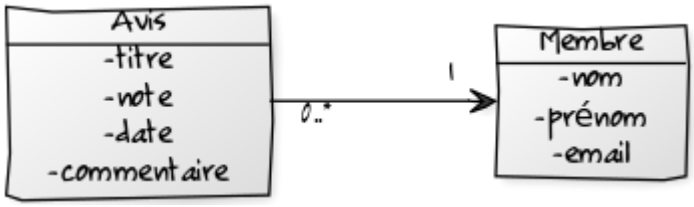
Description du thème

Ce TP est la deuxième partie d'une série sur le Framework Symfony.

Propriétés	Description
Intitulé long	Initiation à Symfony par une suite de TP – Partie 2 : relations entre entités
Formation concernée	BTS Services informatiques aux organisations
Matière	SLAM 4
Présentation	Comprendre les concepts de base de Symfony
Transversalité	SLAM 5
Notions	D4.1 - Conception et réalisation d'une solution applicative Savoir-faire : • Programmer au sein d'un Framework Savoir : • Caractéristiques d'un Framework • Persistance et couche d'accès aux données, technologies et techniques associées • Architectures applicatives : concepts avancés
Pré-requis	Programmation objet Pattern MVC Pattern DAO Serveur web et Symfony installés
Outils	Un environnement de développement pour le web
Mots-clés	Symfony, Framework, PHP
Durée	8h
Auteur(es)	Luc Frébourg
Relectures	Olivier Capuozzo, Gaëlle Castel
Version	v 1.0
Date de publication	Avril 2015

On souhaite gérer les relations entre les différentes classes de l'application TravelAdvisor.

1. En utilisant l'entité Membre, réaliser la classe Avis et la relation correspondante pour que les objets s'enregistrent dans la base. On considère qu'un avis correspond à un Membre. Un membre peut rédiger plusieurs avis ou n'en avoir aucun.

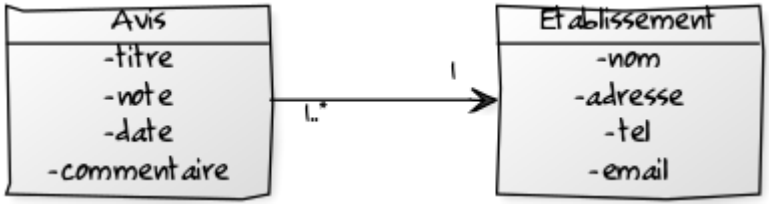


2. Dans une fonction **ajoutAvisAction** appelée depuis **/ajoutAvis**, inclure l'objet membre réalisé à la partie 1 possédant l'**id 1** dans un nouvel avis. On pourra récupérer le membre grâce à un accès à la base de données. On ne les enregistre pas dans la base.

3. Afficher l'avis et le membre en réalisant une vue **ajoutAvis** avec les accès sur les propriétés d'objets grâce à Twig.

4. Enregistrer ces objets dans la base de données dans **ajoutAvisAction**.

5. Réaliser la classe et la relation correspondante pour que les objets s'enregistrent dans la base.



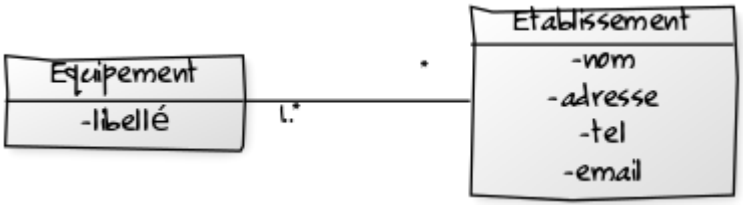
6. Dans une fonction **ajoutEtablissementAction** appelée depuis **/ajoutEtablissement**, enregistrer un établissement avec l'avis **1** dans la base de données.

7. Réaliser la méthode magique PHP _toString() pour chaque entité. On affichera par défaut :
   - Le nom du membre
   - le titre et la note de l'avis
   - le nom et l'adresse de l'établissement

8. Afficher l'avis 1 et les autres entités liées grâce aux méthodes magiques en cascade depuis **/avis**.

9. Réaliser la classe et la relation correspondante pour que les objets s'enregistrent dans la base.

   Une relation bi directionnelle entre les équipements et les établissement est nécessaire. Nous souhaitons accéder facilement à tous les établissements possédant un type d'équipement et la liste des équipements de chaque établissement.



10. Attribuer deux équipements à l'établissement précédent depuis **/etablissement/ajoutEquipement/{libelle}**. On appelera deux fois cette URL. Un message de confirmation avec tous les équipements de l'établissement apparaitra.

11. Ajouter un établissement au dernier équipement depuis **/equipement/ajoutEtablissement/{id}**. Un message de confirmation avec le nom de l'établissement apparaitra.