

Symfony Partie 1

Réalisation d'un bundle avec entité persistée

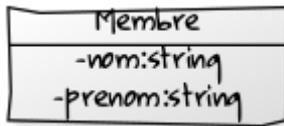
Description du thème

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

Propriétés	Description
Intitulé long	Initiation à Symfony par une suite de TP – Partie 1 : entité persistée
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 : <ul style="list-style-type: none">• Programmer au sein d'un Framework Savoir : <ul style="list-style-type: none">• 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	4h++
Auteur(es)	Luc Frébourg
Relectures	Olivier Capuozzo, Gaëlle Castel
Version	v 1.0
Date de publication	Avril 2015

La société **TravelAdvisor** est spécialisée dans l'évaluation de sites touristiques grâce au recueil d'avis de consommateurs. Pour le moment, l'entreprise possède un ensemble de notations pour deux types d'établissements, des restaurants et des hotels. Chaque membre, une fois inscrit, peut noter les établissements sur cinq et ajouter des commentaires.

Voici la classe Membre :



La société souhaite réaliser un site web dynamique. Pour optimiser la programmation, elle décide d'utiliser le framework Symfony basé sur le langage PHP. Une procédure d'installation est décrite dans le **document joint « Partie 0 installation »**.

Grâce au document de présentation globale (document joint « Partie 1 Présentation globale »), répondre par étape à une partie des besoins de TravelAdvisor. Il est nécessaire de garder ce TP pour réaliser les prochains.

Notation : {var} correspond à une variable

Travail à faire :

1. Réaliser un bundle td/travelAdvisorBundle.
2. Afficher « Bienvenue sur TravelAdvisor » avec l'URL **/voyage**.
3. Grâce à une autre vue, afficher « Bienvenue sur TravelAdvisor *votrePrenom* » avec **/voyage/{votrePrenom}**.
4. Faire en sorte que /voyage/votrePrenom renvoie vers la première vue.
5. Ajouter des balises HTML dans votre vue, <html> et <body> au minimum. Des informations de Symfony apparaissent sur la vue.
6. Définir un deuxième contrôleur qui prendra les URL de la forme **/admin/** . On appellera une vue qui affiche « Espace administrateur ».
7. Réaliser une fonction de redirection qui génère une URL /voyage/visiteur grâce à /admin/redirection.
8. Réaliser une vue qui affiche les avis de 1 à x grâce à une boucle Twig. On se limitera à afficher le numéro. Cette vue sera appelée grâce à **/avis/{x}** . L'emplacement des fonctions est laissé au choix.
9. Modifier le template layout de base app/Resources/views/base.html.twig en ajoutant un pied de page.
10. Faire en sorte que les vues utilisent le template de base.
11. Réaliser un lien vers la première vue depuis la dernière vue réalisée.

Liaison avec Doctrine

12. Générer la classe Membre.
13. Générer la table.
14. Ajouter l'email à Membre dans l'entité et dans la base.
15. Enregistrer un Membre dans la base de données depuis **/ajoutMembre**.
16. Afficher le membre enregistré grâce à une vue depuis **/afficherMembre**.