|  |  |
| --- | --- |
| C:\Users\remi.saint-laurens\Desktop\Aix en Provence\Présentation Y-nov\Y-nov-Logo-RVB.jpg | Module : PHP Framework – B2  Nom de l’intervenant : AZZAG SAMIR |
|  |  |

|  |
| --- |
| Note/ |

|  |  |
| --- | --- |
| Date |  |
| Nom et prénom |  |
| Appréciation \*  \*A remplir par l’intervenant |  |

**Session – 2016/2017**

### Informations

**Durée de l’examen : 2h30**

**Matériel autorisé :** Tous documents autorisés.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Observation du surveillant :**



### QCM 4 pts

**Qu’est-ce qu’un framework ?**

1. Un site Internet prêt à l’emploi
2. Un site Internet à configurer, comme un CMS
3. Une boite à outils à destination des développeurs
4. Une boite noire rigide à utiliser

**Comment qualifier l’utilisation d’un framework ?**

1. Les frameworks sont peu utilisés
2. Les frameworks sont un peu long à apprendre, mais améliorent la productivité ensuite
3. Les frameworks sont trop lourds pour un site web
4. Les frameworks sont trop compliqués à utiliser

**Pour utiliser Symfony3, faut-il obligatoirement maîtriser la console ?**

1. Non, car Symfony3 est un site web.
2. Non, car les commandes sont plutôt simples et toujours facultatives.
3. Oui, car l'utilisation de la console est obligatoire.
4. Oui, car les commandes sont compliquées et il faut de l'expérience.

**Quel est le rôle du contrôleur frontal ?**

1. Il contrôle que le visiteur soit bien identifié sur le site
2. Il contrôle la requête.
3. Il convertit une requête en une réponse grâce au noyau
4. Il instancie la réponse

**Qu'est-ce que l'architecture MVC ?**

1. Une architecture inventée par Symfony3.
2. Une architecture pour organiser son code PHP.
3. Une architecture pour organiser son code quel que soit le langage.
4. Une architecture pour organiser ses méthodes de travail.

**Dans quel langage le composant Console de Symfony est-il écrit ?**

1. En Bash
2. En PHP
3. En Java
4. En C++

**Est-on obligé de passer par la console pour créer un bundle ?**

1. Oui, car elle génère des fichiers impossible à écrire à la main.
2. Oui, car elle enregistre automatiquement le bundle dans le noyau.
3. Non, car on peut faire à la main tout ce qu’elle fait.
4. Non, car elle ne fait rien.

**Quelle commande permet d’envoyer un mail avec symfony 3 ?**

1. php app/console swiftmailer:email:send
2. php bin/console mailer:send
3. php app/console swiftmailer:email:send
4. php bin/console swiftmailer:email:send

### Une application en Symfony 3 16 pts

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 hôtels. Chaque membre, une fois inscrit, peut noter les établissements sur cinq et ajouter des commentaires. Voici la classe Membre :

|  |
| --- |
| **Membre** |
| **- nom:string**  **- prenom:string**  **- created:datetime** |

La société souhaite réaliser un site web dynamique. Pour optimiser la programmation, elle décide d’utiliser le framework Symfony basé sur le langage PHP.

#### Consignes :

Vous trouverez tout le code fait en cours pour l’exemple ynovblog sur le lien suivant : <https://github.com/ynovb2/yblog>

Vous ***répondrez aux questions directement sur ce document***.

Votre rendu contiendra votre code source ainsi que ce document dans une archive au format suivant : ***[DS\_Sym]NOM\_PRENOM.zip***

Vous m’enverrez votre rendu via un lien partage en utilisant une plateforme de votre choix.

#### Travail à faire :

* Quelle commande permet de créer un nouveau projet symfony ?

Php symfony new nomDuProjet

* Quelle commande permet de générer un *Bundle* ?

Php console generate :bundle

*Vous utiliserez le format yml pour les routes, les annotations pour les entités Doctrine et le moteur de template Twig pour les vues.*

* Créer un nouveau projet ynovtravel
* Réaliser un bundle Traveller/TravelAdvisorBundle.
* Expliquez avec vos mots la hiérarchie de dossiers générés par Symfony.

Symfony génère une hiérarchie de dossier comme suit :

Dans le dossier app se trouve tous les fichiers de configuration de notre application

Dans le dossier src se trouve notre projet à proprement parler

Dans celui-ci se trouve le bundle que nous venons de générer le bundle possède un contrôleur qui s’occupe de faire la lisaison entre les données (Entity) et la vue   
envoyé à l’utilisateur. Il possède aussi un dossier ressources dans lequel nous trouvons les templates twig à partir duquel le html final envoyé au client est généré, ainsi que le dossier config dans lequel les routes sont paramétrées.

* Quel dossier est le seul à être accessible par les visiteurs du site en production ?

Le dossier accessible par les visiteurs du site en production est le dossier à la racine /web

* A ce niveau, vous devez partir d’un projet vierge. Détaillez ce que vous devez faire pour cela.

Nous devons commencer par paramétré les routes que l’application utilisera. Créer ou modifier le contrôleur vers lequel elle redirige ainsi que le template twig qui sera utilisé.

* Qu’est-ce que le profiler et comment y accéder ?

Le profiler permet d’obtenir des infos sur toutes les requêtes, le temps de chargement des pages ainsi que la route empruntée par ces dernières. C’est un compagnon indispensable pour débugger une application. Il est accessible en environnement dis de développement grâce au noyau app\_dev.php.

* Afficher « Bienvenue sur TravelAdvisor » avec l’URL ***/voyage***. (La barre d’accès au profiler doit être disponible)
* Grâce à une autre vue, afficher « Bienvenue sur TravelAdvisor *votrePrenom* » avec ***/voyage/{votrePrenom}.***
* Faire en sorte que ***/voyage/votrePrenom*** renvoie vers la première vue.
* Définir un deuxième contrôleur qui prendra les URL de la forme ***/admin/****.* On appellera une vue qui affiche « Espace administrateur ».
* Réaliser une fonction de redirection qui génère une URL ***/voyage/visiteur*** grâce à ***/admin/redirection***.
* Modifier le template layout de base *app/Resources/views/base.html.twig* en ajoutant un pied de page
* Faire en sorte que les vues utilisent le template de base grâce à l’héritage à trois niveaux.
* Quelle commande permet de vider le cache de l’environnement de production ou de développement ?

php bin/console cache:clear --env=prod pour vider l’environnement de production et

php bin/console cache:clear sans préciser l’environnement pour vider l’environnement de développement.

* Générer la classe « Membre » Doctrine avec ses annotations.
* Quelle commande avez-vous utilisé ?

php bin/console generate:doctrine:entity

* Que faut-il faire pour pouvoir utiliser les fixtures Doctrine et mettre en place Doctrine ?

Nous devons installer le bundle fixture en se rendant sur le site packagist

Nous y trouvons la commande : composer require doctrine/doctrine-fixtures-bundle

* Quelle commande permet de mettre en place votre base de données ?

php bin/console doctrine:database:create

* Créez votre base de données
* Générer le schema de la base de données.
* Quelle commande vous a permis de le faire ?

php bin/console doctrine:schema:update --force

* Quelle commande permet de vérifier l’instruction SQL avant son exécution ?

php bin/console doctrine:schema:update --dump-sql

* Quels fichiers avez-vous configuré pour mettre en place votre schema ?

Le fichier/classe Membre.php a été configuré afin de mettre en place le schema.

* Ajouter l’email et le mot de passe à « Membre » dans l’entité et dans la base.
* Quelle commande avez-vous utilisé pour mettre à jour la base et l’entité membre ?

php bin/console doctrine:generate:entities AcmeMyBundle:Customer

Pour mettre à jour l’entité membre

Et php bin/console doctrine:schema:update --force pour mettre à jour la base

* Que sont les fixtures ?
* Les fixtures sont des classes nous permettant de peupler la base de données de données arbitraire.
* Quelle commande permet de peupler(hydrater) la base de données à l’aide de fixtures ?

php bin/console doctrine:fixtures:load

* Peupler(hydratez) la table à l’aide de fixtures (un ou deux membres pour tester).
* Créer un formulaire de connexion ***/connexion*** avec un flashBag qui affichera *‘connexion réussie’* et les validateurs associés aux champs :
  + Login : email
  + Password : minimum 8 caractères
* Comment fait-on pour mettre à jour les liens vers les fichiers css d’un Bundle ?
* Afficher un membre enregistré grâce à une vue depuis***/afficherMembre/{id}***