

Programmation Web

Licence 3 d'informatique 2015-2016

Université Paris Diderot

Examen de session 1 – Durée : 3h
19 mai 2016

1 Modalités de l'examen

Réalisez le mini-projet décrit dans la section 2 en Symfony 3. Le mini-projet de la section 3 en node.js sera, quant à lui, seulement pris en compte comme bonus. Le travail doit être rendu sur Didel, dans la rubrique Travaux – Examen session 1.

L'utilisation de tout moyen de communication est interdite (pas de téléphone, ni e-mail, ni réseaux sociaux, etc.). Vous pouvez en revanche utiliser librement Internet pour vous documenter.

1.1 Rendu attendu

Un fichier `votre_nom_votre_num_etudiant.zip` qui se décompresse dans un répertoire du même nom contenant :

- un document texte nommé `descriptif.txt` décrivant brièvement les fonctionnalités réalisées, à la fois pour le projet Symfony et le projet node.js, le cas échéant ;
- un répertoire nommé `code` contenant votre code. Le répertoire `code` doit contenir :
 - Un répertoire nommé `symfony` contenant les fichiers symfony que vous avez modifiés et/ou ajoutés au framework de base. Cela comprend par exemple les contrôleurs, les vues, les entités, le « repository », les fichiers de configuration, le js, le css, quelques images – pas trop d'images pour ne pas surcharger la soumission. Veillez à bien indiquer dans le document descriptif, le chemin de chaque fichier par rapport au répertoire principal du projet symfony.
 - Un script nommé `bd_sym.sql` dont l'exécution dans `mysql` génère la base de données utilisée par votre projet symfony. Veillez à ne pas insérer trop de données pour ne pas surcharger la soumission.
 - (Si vous soumettez également le mini-projet node.js) Un répertoire nommé `nodejs`, contenant tout votre code node.js ainsi qu'un fichier `package.json`, et les vues nécessaires. Le fichier principal à exécuter doit être nommé `index.js`. Le fichier `package.json` doit contenir dans la section `dependencies` tous les modules node.js utilisés par votre projet,
 - (Si vous soumettez également le mini-projet node.js) un script nommé `bd_node.sql` dont l'exécution dans `mysql` génère la base de données utilisée par le projet node.js.

1.2 Modules externes

L'utilisation de bundles Symfony externes est interdite.

L'utilisation de modules node.js installés depuis `npm` est permise.

1.3 Modalité de test du rendu Symfony

Nous allons tester votre code dans notre environnement Symfony 3 selon la procédure suivante ; veillez à vérifier que cela suffit pour faire fonctionner correctement votre projet. Nous allons :

- créer un projet symfony 3 vierge et y installer vos fichiers selon vos indications ;
- exécuter dans `mysql`, en utilisant notre propre compte, votre script `bd_sym.sql` pour créer la base de données ;
- changer `database_name`, `database_user` et `database_password` dans `app/config/parameters.yml` pour utiliser nos identifiants ;
- lancer la commande `php bin/console server:run` dans le répertoire principal du projet ;
- tester l'application à l'adresse `http://localhost:8000/` (toute l'application doit être accessible depuis la route racine /).

1.4 Modalité de test du rendu node.js

Nous allons tester votre code dans notre environnement node.js selon la procédure suivante ; veillez à vérifier que cela suffit pour faire fonctionner correctement votre projet. Nous allons :

- exécuter dans mysql, en utilisant notre propre compte, votre script `bd_node.sql` pour créer la base de données ;
- changer database, user et password dans la commande `mysql.createConnection` (qui doit se trouver au début du fichier `index.js`), pour utiliser nos identifiants ;
- lancer la commande `npm install` dans le répertoire `nodejs` pour installer les modules requis dans `package.json` ;
- lancer la commande `node index.js` dans le répertoire `nodejs` ;
- tester l'application à l'adresse `http://localhost:8000/` (toute l'application doit être accessible depuis la route racine `/`).

2 Mini-projet Symfony

Réalisez un site complet en Symfony sur un thème de votre choix. Le thème est libre mais doit être centré sur une fonctionnalité de partage, où les utilisateurs peuvent soumettre du contenu accessible aux autres (par exemple un blog, un simple réseau social, un agenda partagé, un site d'annonces, etc.).

Il est interdit de réutiliser des parties de votre projet, un contrôle sera effectué.

Veillez à utiliser tous les outils appris en cours : `css / bootstrap` pour la mise en page, `javascript/jquery` pour l'interaction avec l'utilisateur (html dynamique, validation de formulaires, possiblement communication asynchrone avec le serveur), une base de données pour le stockage du contenu partagé, et Symfony pour le développement serveur.

En vue des contraintes de temps, votre site peut être simple et minimaliste. Il sera noté sur les points suivants :

- utilisation de symfony ;
 - un css minimum mais clair et organisé ;
 - un html cohérent à défaut d'être complet, et prévu pour passer facilement à l'échelle (bonne gestion des classes) ;
 - une version du site utilisable sur petit écran, géré avec Bootstrap ;
 - un peu de JQuery ;
 - fonctionnalité d'enregistrement et de login ;
 - validation de l'input de l'utilisateur dans un formulaire ;
 - une fonctionnalité minimale de recherche (par un critère simple, comme la date ou le titre, etc.).
- D'autres fonctionnalités spécifiques à votre site seront prises en compte comme bonus.

3 Mini-projet Node.js

Réalisez un simple serveur `node.js` pour la gestion d'un formulaire d'enregistrement « à étapes ». Chaque étape, terminée par une soumission de formulaire, demande des informations à l'utilisateur. Selon les données reçues, le serveur établit la prochaine étape, parmi plusieurs possibilités. Par exemple la première étape pourrait demander des informations communes (nom, prénom, emploi, etc.). Ensuite, si l'emploi est de type « salarié », la prochaine page demandera le type entre CDD/CDI, le montant du salaire, etc. En revanche si l'emploi est « indépendant », la prochaine étape demandera le montant des ressources pour l'exercice en cours, etc.

Gérez au moins trois étapes. Le thème est à nouveau libre (l'exemple ci-dessus est donné uniquement à titre indicatif).

La soumission de chaque étape doit être gérée par le serveur, et non pas par le javascript coté client. En revanche (en bonus), n'hésitez pas à ajouter du javascript côté client pour la vérification des champs du formulaire.

Le serveur s'occupe enfin d'enregistrer les données dans une base de données. Concevez le schéma de cette base pour représenter, sans redondances, les données reçues aux différentes étapes.

Vous pouvez utiliser Twig ou Ejs pour les vues.