

# TP4 - Application serveur à l'aide Express/NodeJS

## 27 octobre 2014

### 1 Objectifs

Le but de ce travail pratique, réalisé en équipe de deux personnes exactement, est de vous familiariser avec Express/NodeJS et Jade.

### 2 Introduction

Tout au long du trimestre, vous implémenterez un site web qui génère des questionnaires (quiz) pour les personnes qui désirent tester leurs connaissances sur les notions du Web. Nous nous attarderons principalement sur des questions sur le domaine du HTML5, CSS3 et Javascript, mais nous pouvons étendre ces questions à d'autres technologies du Web telles que HTTP, NodeJS, AngularJS, etc.

### 3 Travail à réaliser

Dans le TP1, le TP2 et le TP3, vous avez élaboré la structure, la mise en forme et la logique de votre site web. Par contre, vous n'avez pas actuellement la possibilité d'effectuer la gestion des formulaires, de communiquer avec une base de données, d'empêcher la duplication de code HTML, etc. Pour ce faire, nous allons ajouter la couche serveur au site.

Au cours du TP4, vous implémenterez la couche serveur du site web à l'aide d'une bibliothèque de NodeJS qui s'appelle Express. De plus, vous utiliserez un préprocesseur nommé Jade pour écrire votre code HTML.

Selon certaines directives, vous serez obligés de modifier votre structure initiale afin de répondre aux exigences.

Pour les personnes qui désirent travailler sur leur portable, voici les instructions à suivre pour installer Express/NodeJS:

1. Il faut tout d'abord installer NodeJS : <http://nodejs.org/>
2. Pour installer Express, il faut lancer la commande `npm install -g express` dans une console. (Il faut sûrement faire `sudo` pour que la commande fonctionne)
3. Ensuite, lancer `express monApp` pour créer votre dossier `monApp` qui contient l'ensemble de votre application serveur Express.
4. Faire `npm install` pour installer les dépendances (avec probablement `sudo`).

5. Exécuter la commande *npm start* lancer votre application serveur au <http://localhost:3000/>.

Le fichier *package.json* contient l'ensemble des modules de dépendances de votre projet. Voici les dépendances que j'ai utilisées (les versions peuvent différer) :

```
"dependencies": {  
  "express": "~4.2.0",  
  "express-session": "~1.7.6",  
  "static-favicon": "~1.0.0",  
  "morgan": "~1.0.0",  
  "cookie-parser": "~1.0.1",  
  "body-parser": "~1.0.0",  
  "debug": "~0.7.4",  
  "jade": "~1.3.0"  
}
```

Chaque fois que vous modifiez le fichier *package.json*, il faut lancer la commande *sudo npm install* pour installer ou modifier les modules de dépendances.

Voici les directives de ce travail pratique :

1. Gestion des routes du site web. À l'aide de Express, vous allez créer les routes de votre application dans le dossier *routes/* situé à la racine du projet. Chaque route est associée à une vue représentée par un fichier Jade.
2. Conversion de tout votre HTML vers le format Jade. Les fichiers *.jade* se trouve dans le dossier *views/*.
3. DRY ou NVRP - Ne vous répétez pas. À l'aide de Express et Jade, aucun bloc de code HTML et Javascript ne doit être dupliqué.

Utilisez les conditions et les boucles de Jade pour éviter la duplication.

4. Création d'un dossier *lib/* dans la racine du projet. Ce dossier contiendra un fichier *db.js*. Ce fichier représente la pseudo base de données du TP3, mais la BD ne sera pas visible à l'extérieur du fichier. Le fichier devrait seulement exporter les fonctions nécessaires pour extraire l'information pertinente de la BD.

Par exemple, une fonction possible: *obtenirQuestionParID(id)*

5. Le gestionnaire de la page du test rapide appelle une fonction de *db.js* pour extraire une question aléatoire. Cette question aléatoire sera affichée à l'aide de Jade.

6. Gestion du formulaire d'initialisation d'un examen du tableau de bord. Le formulaire doit envoyer une requête POST vers la page de question d'examen. Le gestionnaire de cette requête POST va générer les ID des questions d'examens à poser selon les paramètres en entrée du formulaire.

Les domaines, le nombre de questions et les ID des questions choisies seront enregistrés dans la Session de Express. Le gestionnaire POST de la page d'examen va être redirigé vers le gestionnaire GET de la page d'examen. Le gestionnaire GET va récupérer les données de la Session pour les utiliser lors de la génération de la page.

7. Dans les pages de tests rapides et d'examens, il faut utiliser Jade, et non les fonctions du DOM avec JQuery ou JS, pour insérer le titre de la page, le domaine de la question, la question elle-même et ses réponses possibles.

Utilisez les boucles de Jade pour afficher les réponses d'une question.

8. Dans ce TP4, le LocalStorage peut être UNIQUEMENT utilisé pour envoyer la note courante d'un test ou examen de page en page:
  - a. Envoyer la note courante d'un test ou examen entre les pages de question de tests ou d'examen respectivement.
  - b. Garder en mémoire le nombre de questions tests rapides réussies et le nombre de question tests rapides effectuées.
  - c. Garder en mémoire l'ensemble des examens effectués pour remplir la boîte de détails d'examens.
9. L'ensemble des fonctionnalités du TP3 doivent être respectées dans ce TP4.

## 4 Évaluation

Globalement, vous serez évalués sur l'application serveur de votre site qui sera implémentée à l'aide de Express/NodeJS. Plus précisément, le barème d'évaluation est le suivant :

---

Exigences	Points
Gestion des routes	10
NVRP - DRY	20
Gestion du formulaires d'initialisation du examen	20
Gabarits Jade	10
Fonctions de la pseudo base de données	10
Limitation du LocalStorage	15
Respect des fonctionnalités du TP3	15

---

L'évaluation se fera en lançant la commande *npm start* à partir du dossier qui contient app.js (ou autre selon le nom que vous lui aurez donné). À partir de cette page, je devrais être capable de consulter toutes les pages de votre site web.

## 6 Direction de la remise

1. Pour la remise, on demande de placer votre code Express/NodeJS dans un dossier compressé nommé : TP4\_matricule1\_matricule2. Vous avez le choix du type de compression.
2. Vous devrez également créer un fichier nommé temps.txt à l'intérieur du dossier de votre projet. Vous indiquerez le temps passé **au total** pour ce TP.
3. Le travail pratique doit être remis **avant** 23h55 le dimanche 9 novembre 2014 sur Moodle.

## 7 Pénalités

1. La politique pour les travaux pratiques remis en retard: -20% pour chaque jour de retard.
2. Les directives 1 et 2 de la remise: -5% si ces directives ne sont pas respectées.

Bonne chance!