



Contrôle terminal info2 / Semestre 3

M3104 : Développement d'applications Web côté serveur

Nom du responsable :	Nicolas Le Sommer
Date du contrôle :	20 octobre 2020
Durée du contrôle :	1h30
Nombre total de pages :	3
Impression :	Recto-Verso
Documents autorisés :	1 feuille A4 recto/verso de notes personnelles
Calculatrice autorisée :	non
Réponses :	

Dans ce sujet de contrôle, nous considérons une application Web permettant de recenser les machines actives sur un réseau d'entreprise. Dans cette optique, nous supposons que chaque machine adresse à cette application Web une requête HTTP au moment de son démarrage pour annoncer sa présence dans le réseau, requête qui contient le nom de la machine, son adresse IP et son type de système d'exploitation. Nous supposons également que chaque machine adresse une autre requête HTTP à cette application Web lors de son extinction pour signaler sa disparition du réseau ; cette requête contenant uniquement le nom de la machine. Nous supposons également que cette application Web utilise une base de données SQLite pour stocker les informations nécessaires à son fonctionnement.

La requête émise par la machine « deathstar » lors de son démarrage pourra l'être avec la commande suivante :

```
curl -X POST http://192.168.1.1/hosts -d "hostname=deathstar&ip=192.168.1.12&os=linux"
```

La requête émise par la machine « deathstar » lors de son arrêt pourra l'être avec la commande suivante :

```
curl -X DELETE http://192.168.1.1/hosts/deathstar
```

Dans les requêtes précédentes, nous considérons que l'application Web est hébergée par la machine dont l'adresse IP est 192.168.1.1.

Exercice 1 : HTTP (2,5 points)

1. La commande « curl » émet par défaut des requêtes HTTP en utilisant le protocole HTTP 1.1, en se déclarant comme un agent utilisateur « curl » qui accepte tout type de données (i.e., /*/*). Donnez la requête HTTP qui sera émise par la commande suivante :

```
curl -X POST http://192.168.1.1/hosts -d "hostname=deathstar&ip=192.168.1.12&os=linux"
```

Exercice 2 : Base de données (2,5 points)

1. Écrivez le script SQL de création de la base de données utilisée par l'application Web.

Exercice 3 : PHP (7 points)

1. À l'aide de l'API PDO du langage PHP et du modèle DAO (*Data Access Object*), écrivez une classe HostDAO contenant une méthode `findAll()` retournant la liste des machines contenues dans la base de données, dont le script de création est celui de l'exercice 2. Vous considérez la version 7.4 de PHP.
2. Écrivez un script PHP `HostList.php` qui permet d'afficher sous la forme d'un tableau HTML la liste des machines contenues dans la base de données de l'application. Vous utiliserez votre réponse à la question précédente pour répondre à cette question. La page HTML produite devra respecter la norme HTML5. Pour des raisons de simplification, vous n'implanterez pas le modèle MVC (Modèle, Vue, Contrôleur) dans cette question. Vous considérez la version 7.4 de PHP.

Exercice 4 : Javascript/Express (8 points)

En utilisant le cadre de conception Express, NodeJS et le module SQLite3 pour NodeJS, écrivez en Javascript :

1. la fonction chargée de traiter la requête émise par une machine lors de son démarrage ;
2. la fonction chargée de traiter la requête émise par une machine lors de son arrêt.

Ces fonctions seront contenues dans le fichier `routes/hosts.js`. Vous supposerez l'existence des variables `express` et `router` dans ce fichier, variables qui sont initialisées comme suit :

```
var express = require('express');  
var router = express.Router();
```

Ces fonctions devront retourner la chaîne de caractères « error » si une erreur intervient lors de l'ajout ou du retrait de la machine dans la liste des machines actives sur le réseau, et la chaîne de caractères « success » si les opérations se sont bien déroulées.

Vous supposerez également que ces fonctions seront utilisées par l'application Express grâce aux instructions suivantes définies dans le fichier `app.js` de l'application Web.

```
var hostsRouter = require('./routes/hosts');  
var app = express();  
app.use('/hosts', hostsRouter);
```