

INF4375 – Énoncé du TP2
À remettre le 23 juin 2014 à 17h30

Lors du TP1, vous avez construit une base de données contenant des dossiers d'étudiants et des groupes-cours. Cette fois, vous devez concevoir le backend d'une application web qui exploitera cette base de données. Ce travail doit être fait individuellement.

Fonctionnalités

Vous devez concevoir un backend qui offre les services REST suivants :

1 – Consultation d'un dossier d'étudiant

Description : Envoie au client le dossier complet de l'étudiant, en format JSON.

Méthode : GET

URL : /dossiers/:cp (où cp est le code permanent de l'étudiant)

2 – Ajout d'un dossier d'étudiant

Description : Reçoit du client un dossier complet d'étudiant, en format JSON, et crée le dossier. Le document JSON est encodé dans le body de la requête HTTP. La structure de l'objet doit être la même que celle stockée dans mongodb, à l'exception de la propriété `_id` qui ne doit pas être présente. Si la structure n'est pas la bonne, la requête est rejetée.

Méthode : POST

URL : /dossiers

3 – Modification d'un dossier d'étudiant

Description : Reçoit du client l'ensemble des modifications à apporter au dossier, en format JSON, et les applique au dossier. Le document JSON est encodé dans le body de la requête HTTP.

Méthode : PUT

URL : /dossiers/:cp (où cp est le code permanent de l'étudiant)

4 – Suppression d'un dossier d'étudiant

Description : Supprime le dossier de l'étudiant. Il est impossible de supprimer un dossier si l'étudiant a déjà complété un cours avec succès.

Méthode : DELETE

URL : /dossiers/:cp (où cp est le code permanent de l'étudiant)

5 – Consultation d'un groupe-cours

Description : Envoie au client les données d'un groupe-cours, en format JSON.

Méthode : GET

URL : /groupes/:oid (où oid est l'ObjectId du groupe)

6 – Ajout d'un groupe-cours

Description : Reçoit du client les données complètes d'un groupe-cours, en format JSON, et crée le groupe-cours. Le document JSON est encodé dans le body de la requête HTTP. La structure de l'objet doit être la même que celle stockée dans mongodb, à l'exception de la propriété `_id` qui ne doit pas être présente. Si la structure n'est pas la bonne, la requête est rejetée.

Méthode : POST

URL : /groupes

7 – Modification d'un groupe-cours

Description : Reçoit du client l'ensemble des modifications à apporter au groupe-cours, en format JSON, et les applique au groupe-cours. Le document JSON est encodé dans le body de la requête

HTTP.

Méthode : PUT

URL : /groupes/:oid (où oid est l'ObjectId du groupe)

8 – Suppression d'un groupe-cours

Description : Supprime le groupe-cours. Il est impossible de supprimer un groupe-cours des étudiants y sont inscrits.

Méthode : DELETE

URL : /groupes/:oid (où oid est l'ObjectId du groupe)

Considérations techniques

Vous devez utiliser la version 2.4.5 de MongoDB ou une version plus récente. Votre base de données doit être nommée selon votre code permanent.

Vous devez utiliser la version 0.9.12 de Node.js ou une version plus récente. Vous pouvez utiliser tous les modules npm que vous voulez.

Le projet doit être fait avec le framework Express.js. Peu importe la version d'express utilisée, les numéros de version d'Express.js et Jade doivent être explicitement indiqués dans le package.json.

Les fichiers du TP1 doivent être remis avec le TP2. Vous allez créer un répertoire "migration" à la racine de votre projet et y placer les fichiers du TP1 (migration.js, son package.json et les deux fichiers XML).

Vous devez fournir une documentation expliquant comment invoquer vos services, ce qu'ils prennent en entrée et ce qu'ils fournissent en sortie. Vous devez mettre un accent sur le format des documents JSON échangés pour chaque service. La documentation doit être livrée en format PDF, HTML ou markdown.

Remise du travail

Le travail doit être remis à l'aide du logiciel Moodle au plus tard le 23 juin à 17h30. Les travaux en retard ne seront pas acceptés. Vous allez remettre votre projet dans un dossier compressé (zip). Le nom du fichier zip doit contenir votre code permanent. Le zip remis ne doit pas contenir de dossier node_modules. Vous serez pénalisés si vous ne respectez pas ces exigences.

Le projet doit contenir un fichier README indiquant comment construire votre base de données, exécuter votre application et accéder à la documentation des services.

Pondération

Fonctionnement des services – 40%

Respect des principes REST – 30%

Utilisation de MongoDB – 10%

Qualité du code – 10%

Documentation – 10%