Travail sommatif 1

Ce travail est sommatif

Développement de services d’échange de données

420-5SE-AAObjectifs

**Validation** des éléments de compétence « Préparer l’environnement de développement informatique », « Programmer la logique applicative du service » et « Programmer une application de test utilisant le service ».

Pour réussir ce travail, vous devez être capable de :

* Exécuter des requêtes http à l’aide de Postman
* Développer une application web simple utilisant un service d’échange de données
* Exécuter des requêtes asynchrones en javascript depuis le navigateur
* Accéder à un service d’échange de données en Nodejs
* Installer un paquetage à l’aide de npm
* Utiliser des modules en javascript dans l’environnement Nodejs
* Utiliser Express pour construire un api REST
* Déclarer des routes avec Express
* Organiser adéquatement en modules le code javascript côté serveur
* Faire correspondre les méthodes HTTP avec les opérations de persistance des données
* Choisir le bon code d’état lors d’une réponse HTTP

# Réalisation

**En équipe de deux.**

En javascript, codifiez le service d’échange de données fournissant les données d’enregistrements d’une base de données :

* Choisissez et définissez un schéma de données à échanger depuis votre service d’échange de données (ex : vidéothèque, joueurs de hockey, etc);
  + Le schéma doit avoir un nombre restreint de tables (de 1 à 3 environ);
  + Au moins 5 champs doivent être présents;
  + Ajoutez quelques milliers de données de tests;
* Codifiez le service d’échange de données en fournissant des routes pour les opérations suivantes :
  + Obtenir une liste des données, ayant comme paramètres;
    - (Optionnel) Triée sur un champ, par défaut la clé primaire;
    - (Obligatoire) Limitée par page (ex : obtenir la page 1 retourne les éléments de 1 à 20, la page 2 les éléments de 21 à 40, etc.);
    - (Optionnel) Le nombre par page, par défaut 50. Maximum 100 et minimum 10;
    - Limitez le nombre de champs retournés aux quelques champs affichés dans la page web dans la vue liste;
  + Obtenir un seul enregistrement selon un identificateur unique;
  + Ajouter un enregistrement;
  + Modifier un enregistrement;
  + Supprimer un enregistrement;
  + Télécharger un document « pdf » des données de la page actuelle;
* Les données sont présentées au client depuis le service d’échange de données dans le format de données JSON.

Codifiez des requêtes HTTP dans Postman pour tester votre service d’échange de données.

Codifiez une application web pour tester votre service d’échange de données.

* Les enregistrements sont affichées en vues « liste-détails »;
* La liste des enregistrements présente des informations limitées (seulement quelques champs, par exemple nom et prénom);
* Les enregistrements sont mises-à-jour et obtenues avec des requêtes « XHR » (ou bien l’api fetch);
* L’utilisateur peut
  + Afficher la liste des enregistrements par page;
  + Afficher les détails d’un enregistrement de la liste en cliquant dessus;
  + Ajouter un enregistrement (à partir de la vue détails);
  + Modifier un enregistrement (à partir de la vue détail);
  + Supprimer un enregistrement (à partir de la vue liste);
* L’application est simple et fonctionnelle (javascript vanille seulement).

**Note** :

Hormis dans le code client HTML, vous pouvez utiliser n’importe quelle librairie sur « npmjs ». Vous pouvez également utiliser n’importe quelle version de javascript (ES5, ES6, etc), côté serveur ou client. Soyez en mesure de justifier vos choix si je juge nécessaire de vérifier.

# Critères d’évaluation

Ce travail compte pour **30% de la session**.

La correction est basée sur les observations suivantes :

Livraison des fichiers en conformité avec les modalités imposées

* Les fichiers sont correctement archivés (zip)
* Le nom de l'archive respecte la nomenclature imposée
* Seulement les fichiers nécessaires et tous les fichiers sont présents
* Le nom des fichiers respecte les noms imposés
* Les fichiers sont placés aux bons endroits

Application rigoureuse des consignes de réalisation

* Le schéma démontre la complexité requise
  + Les données de tests sont insérées dans la base de données
  + Une ou plusieurs tables sont présentes (1 à 3)
  + Au moins 5 champs
  + Les requêtes à la base de données respectent les standards de programmation
  + Les requêtes sont effectuées en respectant les standards de sécurité
* Les opérations du service d’échange de données sont fonctionnelles
  + La logique applicative est implantée correctement
  + Le programme ne plante pas
  + Aucune pile des appels ne fuie vers le client
  + Les requêtes aux ressources sont asynchrones
* L’api du service d’échange de données respecte le principe REST
  + Les paramètres de requête sont utilisés adéquatement
  + Les routes sont correctement déclarées
  + Les codes d’état retournés sont valides
* L’application HTML est fonctionnelle et intuitive
  + Les bonnes ressources sont obtenues à partir du service d’échange de données
  + L’application n’utilise que du javascript vanille
  + Les requêtes sont asynchrones en « XHR » ou « fetch »
  + L’interface graphique est simple
  + Le contenu de la page est facilement lisible
  + Les messages d’erreurs sont significatifs

Rédaction de requêtes de test avec Postman

* Une collection regroupe toutes les requêtes
* Chaque route possède une requête
* Les code d’états sont également testés
* Les requêtes sont clairement nommées

Participation au développement du service d’échange de données et de l’application

Possiblement, pour vérifier votre participation au développement du SÉD, je peux vous convoquer individuellement pour observer les éléments suivants :

* Démontrer une bonne compréhension du système dans son ensemble
* Vulgariser le fonctionnement du service d’échange de données et de l’application
* Décrire en détails les parties individuellement développées
* Présenter clairement les problèmes et solutions rencontrés
* Répondre adéquatement aux questions de l’enseignant