



Formation Javascript

LP API-DAE

TP noté

Pierre Cavalet (pcavalet@kaliop.com)

Baptiste Leulliette (bleulliette@kaliop.com)

Sujet du TP :

Il s'agira de développer deux parties:

- La première partie sera une API qui possède X routes. (TODO Pierre)
- La seconde partie sera sur la partie front, en VueJS, qui consomme l'API précédemment développée afin de pouvoir utiliser convenablement votre TODO-list.

Nous attendons un repository github dans lequel nous aurons deux dossiers :

- Un dossier **frontend** qui contiendra tout le code de la partie VueJS
- Un dossier **backend** qui contiendra le code de l'API NodeJS

Un Readme d'explication et d'installation serait un plus sur ce qui a été développée.

Partie Backend

L'API Rest en NodeJS

Rendu Attendu :

- Un readme.md détaillant ce qui a été fait ainsi que les difficultés rencontrées
- Une organisation propre avec des modules

Fonctionnalités attendues :

Une API de gestion de TODO-list en session. Chaque utilisateur qui utilise l'API recevra un cookie permettant de l'identifier, et sa TODO-list sera gérée dans sa session.

La gestion des TODOs sera faite dans un module dédié.

Il est attendu d'utiliser:

- Un middleware qui permet de gérer l'initialisation de la TODO-list. (si elle n'est pas créée, on la crée en session).
- Le middleware d'express vu en TP pour gérer les sessions

Chaque route devra définir dans ses headers le type de contenu renvoyé via le header "Content-Type".

Il est attendu d'avoir:

- Une route permettant d'ajouter un TODO
 - Cette route devra gérer la soumission de formulaire x-www-urlencoded comme vu en TP avec le middleware body-parser
- Une route permettant de supprimer un TODO
- Une route permettant de récupérer ses TODOs selon une pagination, en lui envoyant la 'offset et size' (le nombre de todos par page).

Partie front - La Single Page Application - VueJS

Rendu Attendu :

- Un readme.md détaillant ce qui a été fait ainsi que les difficultés rencontrées
- Une architecture en composants cohérente en fonction des différentes fonctionnalités du TP noté
- Le plus de fonctionnalités possibles parmi la liste ci-dessous

Fonctionnalités attendues :

L'affichage va se découper en plusieurs parties :

- La première partie concerne la liste des tâches restantes dans votre TODO-list
La liste des tâches devra être le point d'entrée des autres actions
L'affichage de cette liste sera affiché sur la première page
- Sur la page d'accueil (donc votre liste des tâches), vous ajouterez un bouton "Créer une tâche" qui permettra de rajouter une tâche à faire sur cette liste. Il s'agira de se déplacer vers un autre affichage (une autre route) qui contiendra un formulaire de création d'une tâche. Une fois que le serveur a validé votre nouvelle tâche, il faudra que votre liste des tâches à faire de la page d'accueil se mette à jour.
- Sur la liste des tâches de la page d'accueil, vous ajouterez une "croix" cliquable qui permettra de supprimer une tâche auprès du serveur.
- Toujours sur cette liste de la page, vous ajouterez un bouton qui permet de voir en détail la tâche qui a été sélectionnée.
- Pour finir, une dernière action sera possible, dans la liste des tâches, afin de modifier une tâche. Il s'agira d'utiliser la même page que la page de création afin de pré-remplir les champs de la tâche en cours de modification et de sauvegarder les modifications auprès du serveur.

Bonus :

Une pagination de la liste des tâches (6 tâches par page) est un plus si l'ensemble des fonctionnalités attendues sont présentes **et fonctionnent correctement**.