

Exercice récapitulatif

Vous devez réaliser une application capable de calculer la moyenne des notes d'une épreuve. L'utilisateur pourra saisir les notes des étudiants et en obtenir la moyenne. Le framework Backbone.js vous a été imposé, ainsi que handlebar et les spécifications suivantes sont à respecter.

A) Un « Model » pour les notes

Créez un « Model » nommé ModelNote pour la représentation des notes. Une note d'étudiant possède 3 attributs :

« nom », « prenom », « note ». Ce « Model » doit fixer la valeur par défaut de « note » à 1. Une méthode « validate » doit être présente, elle doit vérifier que la « note » est bien un **nombre** entre **1** et **6** (c.à.d. dans l'intervalle **[1, 6]**), que le nom est bien une chaîne de caractère non vide, et que le prénom est bien une chaîne de caractère non vide. Il vous est conseillé de tester votre « Model » en créant une instance et en la validant.

B) Une « collection » de notes

Créez une collection nommée ModelNotes dont les « Models » seront des instances de ModelNote. Cette collection devra être en tout temps triée par ordre décroissant des valeurs des notes. Il vous est conseillé de tester votre code en créant une collection, puis d'y ajouter quelques notes afin de vérifier (avec un console.log par exemple) que celles-ci sont bien triées.

C) Templates

Créez les « templates » nécessaire à l'affichage des notes. Pour chaque note, vous devez afficher le nom et le prénom de l'étudiant à laquelle elle est rattachée et sa valeur. De plus 3 boutons par note seront disponible, le 1^{er} supprimera la note, le 2^e permettra sa modification, le 3^e permettra de comparer la note à la moyenne des notes. Vous devez aussi réaliser le formulaire permettant la création des notes. Finalement, votre liste de notes affichera la moyenne de toutes les notes en entête à la liste. La liste des notes doit être triée par **ordre décroissant**. Utiliser le framework bootstrap pour la mise en forme graphique.

D) Création des « Views »

Créez les « views » nécessaires à l'affichage des notes, puis gérez les événements d'ajout, de suppression et de comparaison comme décrit dans les points suivants.

E) Gestion des événements d'ajout d'une note

Dans la bonne vue, gérez le click de souris sur le bouton « ajouter » . Lors d'un click, vous devrez récupérer les valeurs des trois champs du formulaire, puis créer une nouvelle note avec ces trois valeurs. De plus, vous devez valider cette note. Si la note n'est pas valide (champ vide, ou note incorrecte) vous devez afficher une erreur avec un composant « Modal » de Bootstrap et ne rien faire d'autre. La validation **doit se faire** avec la méthode Backbone de validation que vous avez codée dans **ModelNote** au point A). Si la note est valide, celle-ci doit être ajoutée à la collection de notes.

F) Gestion de la suppression d'une note

Dans la bonne vue, gérez les clicks de souris sur les boutons « supprimer». Lors d'un click sur un de ces boutons, la note associée doit être supprimée.

G) Modification d'une note

Dans la bonne vue, gérez les clicks de souris sur les boutons « modifier» . Lors d'un click sur un de ces boutons, un formulaire de modification doit apparaître, avec les valeurs de la note dans les bons champs.

H) Comparaison d'une note

Dans la bonne vue, gérez les clicks de souris sur les boutons « comparer». Lors d'un click sur ces boutons, la note associée doit être comparée avec la moyenne. Pour ce faire, calculer la différence de la note avec la moyenne des notes. Finalement, afficher simplement cette différence avec un composant « modal » de Bootstrap.