TP8 VueJS

Objectifs

Utiliser Vue/cli pour créer des projets.

Comprendre le mécanisme de route avec VueJS

Utiliser les composants « VueJS » en comprenant la communication parent/enfant.

Utiliser Vuex pour gérer des données inter-composants.

Récupérer les données d'une API.

Durée

3h

Préparation du poste de travail

Vous pouvez utiliser vos machines personnelles pour réaliser ce TP ou utiliser les machines virtuelles de l'Université.

Quiz

Le but de cet exercice est de réaliser un QCM sur différentes catégories.

- En utilisant un terminal, créez un projet VueJS appelé quizs. En sélectionnant manuellement les options, ajoutez Vuex ainsi que le routeur. Attention, il faut utiliser la version 2 de VueJs Utilisez les options par défaut en choisissant npm comme gestionnaire de dépendances.
- Ouvrez le projet avec WebStorm.
- Lancez le projet en tapant npm run serve et observez le résultat.
- Les données sont situées dans un fichier json disponible sur moodle.
- Dans le fichier store.js, copiez/collez le contenu du fichier json afin de créer une variable quizs dans l'objet state. Vérifiez que vous voyez ces données dans Vue DevTools (Vuex) au niveau de votre navigateur.
- Renommez le composant HelloWorld en Quizs et videz tout le code superflu.

- Ce composant aura pour but d'afficher les catégories des questions disponibles depuis le store.
    - Créez une variable calculée (computed) quizs qui sera initialisée par le contenu du store.
    - Faites une boucle afin d'afficher chaque catégorie
- Ajoutez une route paramétrée qui permettra d'appeler un composant Quiz. Le paramètre devra permettre de choisir la catégorie de questions.
- Créez le composant Quiz qui pour l'instant affiche simplement un titre. Vérifiez que votre route fonctionne.
- Modifiez le composant Quizs afin de faire en sorte que chaque catégorie envoie vers la route quiz associée à travers un clic.
- Dans le composant Quiz, créez deux variables calculées permettant de récupérer le nom de la catégorie ainsi que les questions associées. Affichez le nom de la catégorie à la place du titre. On doit pouvoir afficher un message d'erreur si le numéro de la catégorie ne correspond à rien.
- Affichez la première question avec les réponses possibles
- Modifiez le code afin de coller le plus possible au modèle : https://ntrugeon.lpmiaw.univ-lr.fr/vuequiz/

Quiz avec API

- Sur les machines virtuelles est installé un petit logiciel appelé json-server qui permet de transformer un simple fichier json en une API restfull.
  Vous pouvez facilement l'installer chez vous avec la commande npm :
  npm install -g json-server
- Dans webstorm, ouvrez un nouveau terminal et lancez la commande json-server db.json. Cela crée une API fonctionnelle sur le port 3000 de votre ordinateur. Vous pouvez tester votre api avec votre navigateur internet et en tapant une des urls fournies.
- Ajoutez Axios à votre projet.
- L'objectif est de modifier le store afin que les données du quiz soient issues de l'API en commentant les données en dur et en remplaçant la variable quizs par un tableau vide. Pour cela, vous devez implémenter les mutations et les actions du store.
    - Dans le fichier store/index.js, importez axios comme indiqué dans le cours.

- Vous allez maintenant ajouter une action et une mutation :
  L'action permet d'appeler de manière asynchrone l'API :

```
getData(context) {
  Axios.get('http://localhost:3000/quizs')
      .then(response => response.data )
      .then( donneesQuiz => {
        context.commit("setData", donneesQuiz)
      })
}
```

- Vous allez maintenant ajouter une mutation qui aura pour rôle d'initialiser le state

```
setData(state, data) {
  state.quizs = data
}
```

- C'est la fonction getData qui appelle setData qui initialise la variable quizs du state.
- Il reste plus qu'à appeler l'action getData. Pour ce faire, vous pouvez appeler cette fonction à travers la méthode dispatch dans l'objet created du composant Quizs.vue.

Pour résumer, lors de la création du composant Quizs.vue, l'action du store est lancé. Cela appelle l'API puis acte les données reçues en initialisant une variable du store.