

TP9 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.

Application météo

Le but de cet exercice est de réaliser une application météo capable d'afficher la météo sur une ville donnée.

- En utilisant un terminal, créez un projet VueJS appelé meteo. En sélectionnant manuellement les options, ajoutez Vuex ainsi que le routeur. Utilisez les options par défaut en choisissant npm comme gestionnaire de dépendances.
- Ajoutez Axios à votre projet.
- Ouvrez le projet avec WebStorm.
- Lancez le projet en tapant npm run serve et observez le résultat.
- Créez un compte sur le site openweathermap.org. Cela vous permet d'obtenir une clé d'API permettant d'interroger les données du site.
- Utilisez votre clé d'API et la documentation afin d'obtenir l'URL des données de météo du jour de la ville de La Rochelle et de la ville de Nice.

Pour la suite, vous pouvez mettre les options de la requête dans l'ordre que vous voulez pour construire plus facilement votre requête.

- Dans le fichier store/index.js, placez votre clé d'API ainsi que l'URL de base :
`cle: 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx',`
`urlMeteo: 'https://api.openweathermap.org/data/2.5/weather?`
`lang=fr&units=metric&q=',`
- Renommez le composant HelloWorld en Meteo et videz tout le code superflu.
- Le composant Home parent contiendra dans un premier temps une variable ville qui est initialisé à « La Rochelle »
- Le composant Meteo aura pour but d'afficher la météo d'une ville.
 - Cette ville sera passée en paramètre du composant
 - C'est ce composant qui fait l'appel à l'API et permet de récupérer la température actuelle, les coordonnées GPS de la ville, et les conditions météo. Cet appel sera fait dans la fonction mounted.
 - Afficher les données de manière pertinente.

Améliorations

- Remplacez la variable ville par un tableau villes permettant de stocker plusieurs villes.
- Un bouton « ajouter » permettra d'ajouter une ville à ce tableau à l'aide d'un composant Config qui affichera un formulaire permettant de saisir la ville et un bouton sauvegarder.
- Modifiez le composant Home afin de boucler sur villes pour afficher plusieurs composants Meteo.
- En utilisant le LocalStorage, mémorisez les villes ajoutées par l'utilisateur. Cela permettra d'afficher dès le lancement de l'application, les villes enregistrées.

Bonus

Une api Symfony est disponible à cette adresse :

<https://ntrugeon.lpmiaw.univ-lr.fr/apivillesymfony/public/api/cities>

Affichez dans le composant Meteo la liste des noms des villes de l'API ainsi que leurs coordonnées GPS. Faites un console.log des éléments reçus afin de comprendre les objets reçus.

Cela vous permettra de mieux aborder la suite de votre projet.

Démo

Vous pouvez voir un exemple de ce qu'il est possible de faire avec cette API.

<https://ntrugeon.lpmiaw.univ-lr.fr/vuemeteo/>

Dans cet exemple, on a rajouté un composant leaflet permettant de visualiser une carte pointant sur la ville (merci aux coordonnées GPS).

Une seconde API gérée par Symfony (celle du dessus) est utilisé pour ajouter facilement les villes depuis le formulaire à l'aide d'un datalist.

Vous pouvez également choisir l'unité des températures entre °C et °F.