

TP7 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.

Durée

3h

Préparation du poste de travail

Vous pouvez utiliser **vos machines personnelles** pour réaliser ce TP.

Vous allez devoir installer npm pour travailler : <https://nodejs.org/en/download/>

La version LTS (Long Term Support) est recommandée.

Installez également vue-cli : <https://cli.vuejs.org/guide/installation.html>

L'option -g permet d'installer de manière globale le framework. Pour cela, vous devrez sûrement avoir des droits d'administrateur sur votre machine.

Installez l'extension VueJS pour votre navigateur.

Galerie 2

Le but de cet exercice est de refaire la mini galerie du TP précédent en utilisant les composants.

- En utilisant un terminal et la commande vue create, créez un projet VueJS appelé garlerie2. Utilisez les options par défaut (version 2 de VueJs) en choisissant npm comme gestionnaire de dépendances.
- Ouvrez le projet avec WebStorm.
- Lancez le projet en tapant npm run serve et observez le résultat.
- Copiez les dossiers « images » et « css » dans le dossier src/assets du projet.
- Nous voulons découper notre application en trois composants :

- Le composant « Galeries » qui affichera la ou les galeries (un ou plusieurs composant Galerie). Vous renommerez le composant « HelloWorld » en « Galeries ». Ce composant contiendra les données des images.
 - Le composant « Galerie » qui affichera les miniatures.
Il contiendra une méthode `changePhoto` et une méthode `imageUrl` qui permet gérer les assets comme dans symfony.

```
imageUrl(p) {  
  // permet de placer les images dans le dossier assets et que cela fonctionne  
  // qu'elle que soit la route...  
  return require('@assets/' + p)  
}
```
 - Le composant « ImageView » contiendra les données de titre et d'image. Il affichera le titre et l'image principale. Il contiendra le code HTML de la section photo. Les données seront passés en paramètres via props.
- La feuille de style css sera importée dans le fichier `main.js` par un « import ».
 - En utilisant la communication parent/enfant, faites fonctionner cette nouvelle version de la galerie.
 - Si vous avez bien structuré votre code, vous devez être capable d'appeler une seconde fois le composant « Galerie » avec de nouvelles données comme dans la vidéo de présentation de VueJS.

Carnet d'adresses

Créez un nouveau projet appelé `vue7` en sélectionnant les éléments manuellement et en ajoutant le mécanisme de routage. Nous allons également utiliser une autre dépendance pour nous faciliter la vie sur la mise en page du site : `materialize`.

```
npm install materialize-css
```

Vous trouverez à cette adresse un exemple de rendu qu'on peut faire avec `materialize` :

<https://ntrugeon.lpmiaw.univ-lr.fr/vue7/>

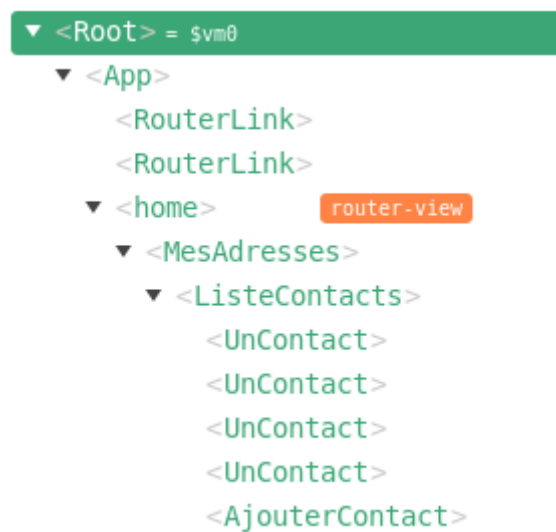
- Importez les css et le js en ajoutant dans le fichier `main.js` les deux lignes suivantes :

```
import 'materialize-css'  
import 'materialize-css/dist/css/materialize.css'
```
- Renommez le composant « HelloWorld » en « MesAdresses ».
- La notation des composants en VueJS peut se faire de différentes manières. Si on veut que le nom de nos composants soit compatible avec le DOM, il faut utiliser le kebab-case à la place du PascalCase lors de l'appel de celui-ci.

Si je crée un composant MesAdresses, il faudra que je l'appelle avec la notation <mes-adresses> dans le template.

- Le composant « MesAdresses » possédera un tableau de personnes composées de ses noms, prénoms, adresses et numéros de téléphone ainsi qu'un booléen favori. Il affichera un composant ListeContacts
- Le composant « ListeContacts » aura pour rôle d'afficher sous forme de liste les contacts favoris et de pouvoir en ajouter.
- Pour cela, nous aurons un composant « UnContact » et un composant « AjouterContact » qui contiendra le formulaire d'ajout.

Voici l'architecture du projet :



- Construisez votre projet en faisant un npm run build et regardez le résultat dans le dossier dist.