

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Le framework Vue.JS

LP CréaWeb

Nicolas PARIS

IUT de Lens – Université d'Artois

Décembre 2021



Ce cours se basera également sur les supports suivants
(marqués par 🔍) :

- <https://v3.vuejs.org/>
- slides from Jadson Santos (Computing Engineer)



Vue.js (ou Vue), qu'est ce que c'est ?

- Framework JavaScript front end open-source...
- ... flexible et léger, il permet de prototyper rapidement des interfaces utilisateur...
- ... et le développement d'applications fluides et réactives appelées Single Page Application (application web monopage)

Définition

Une SPA est une application web accessible via une page web unique : au lieu de naviguer entre différentes pages, le contenu de cette page change dynamiquement et passe donc par différents états durant la navigation

Mais aussi ?

- Il permet le data binding : les données et le DOM sont couplés et réactifs aux changements
- Il incorpore le concept de virtual DOM : Le DOM n'est pas directement changé, ça passe par le virtual DOM
- Il s'organise en composants : une application est souvent découpée en plusieurs sous-composants qui gèrent chacun leur vie et qui sont réutilisables

Rappel

Le DOM est une représentation du document HTML source sous la forme d'un arbre dont l'élément racine est `<html>` et un élément est par exemple `<head></head>`

Remarque

Vue ne supporte pas IE8 et les versions antérieures, car il utilise des fonctionnalités ECMAScript 5 qui ne peuvent pas être émulées sur IE8. Cela dit, Vue supporte tous les navigateurs compatibles ECMAScript 5



Vue, comment on l'installe? 🔍

Via téléchargement manuel

- Télécharger `vue.js` depuis le site <http://vuejs.org>
- Placer ce fichier dans le répertoire de votre site web et chargez-le dans votre page HTML

1

```
<script src="vue.js"></script>
```

Via l'inclusion d'un CDN (le plus simple)

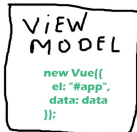
- Charger directement cette librairie depuis un serveur distant via le CDN (Content Delivery Network) suivant :

1

```
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
```

Comment ça fonctionne ? il s'agit du modèle MVVM

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Hello World - Vue.js</title>
  <script type="text/javascript" src="js/vue.js"></s
</head>
<body>
  <div id="app"> {{ message }} </div>
  <script type="text/javascript">
    new Vue({
      el: "#app",
      data: {
        message: "Hello World!"
      }
    });
  </script>
</body>
</html>
```



Le DOM (View) est lié via data-binding avec notre instance de vue (View Model), dans laquelle on trouve la data (Model)

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Instances de vue et cycle de vie 🔍

=> <https://fr.vuejs.org/v2/guide/instance.html>

=> <https://v3.vuejs.org/api/options-lifecycle-hooks.html>

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Exercice 1

Créer un premier HelloWorld en Vue.js

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Les directives

Vue est livré avec des directives intégrées que vous trouverez utiles, chacune ayant son propre cas d'utilisation

Une directive permet d'interagir avec des éléments HTML d'une page, en leur attachant un comportement spécifique



Quelques exemples de directives

- `v-text` : modifie le contenu texte
- `v-html` : modifie le contenu html
- `v-if` : afficher un élément selon une condition
- `v-show` : masquer ou faire apparaître un élément
- `v-else` : à utiliser avec `v-if` et `v-show`
- `v-on` : permet de mettre en place un événement
- `v-bind` : permet de créer une liaison
- `v-model` : lier la valeur d'un contrôle de formulaire
- `v-for` : crée un objet vue.js pour chaque élément se trouvant dans un tableau ou un objet
- `v-el` : crée une référence d'un élément du DOM

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Entrons un peu plus dans le détail 🔍
=> <https://v3.vuejs.org/api/directives.html#v-text>

Exercice 2

Quelques exercices sur les directives

- `file:///home/nicolas/Documents/Enseignement/Vue/10-13122021/exercices/CDN/exvif.html`
- `file:///home/nicolas/Documents/Enseignement/Vue/10-13122021/exercices/CDN/exvmodel.html`
- `file:///home/nicolas/Documents/Enseignement/Vue/10-13122021/exercices/CDN/exvfor.html`
- `file:///home/nicolas/Documents/Enseignement/Vue/10-13122021/exercices/CDN/exvforif.html`

Exercice 3

Le but de cet exercice est de créer un champ texte avec un nombre de caractères limité à la saisie. Le nombre de caractères restants est affiché et prend la couleur rouge lorsqu'il passe à 0 ou moins

indice

- Utiliser la directive `bind:class` pour l'affichage rouge
- Prévoir une variable contenant le nombre de caractères autorisés à la saisie
- Utiliser une propriété calculée (computed) pour le compteur (<https://fr.vuejs.org/v2/guide/computed.html>)

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier


Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Directives de gestion des evenements 
=> <https://fr.vuejs.org/v2/guide/events.html>
(Reprendre code du bouton et tester)

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Exercice 4

Même exercice que le précédent, sauf que maintenant au moindre caractère saisi en trop, une popup d'alerte est proposée

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Les composants

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Vue.js en mode Application de production 🔍

Via Vue-CLI

- Vue met à disposition un CLI (Command Line Interface) pour démarrer rapidement un projet de développement d'application SPA

```
1 npm install vue-cli -g
```

- Pour créer le nouveau projet, on utilise webpack

```
1 vue init webpack monprojet
```

- Le nouveau projet est prêt à être lancé

```
1 cd monprojet / npm install / npm run dev
```

- L'environnement est déjà en place et une page est déjà prête sur `http://localhost:8080`

Structure projet Vue

- package.json : les informations de package NPM du projet (dépendances, versions,,...)
- src : les sources de votre projet
- src/assets : des ressources statiques (images, fichiers)
- src/components : vos composants Vue
- src/App.vue : votre composant Vue racine, qui contient toute l'application

index.html

Le fichier index.html est quand à lui le point d'entrée de l'application. Il comporte uniquement le code html de base ainsi qu'une div avec App (le composant racine) qui nous servira de point de départ du code de l'application

```
<> index.html x
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width,initial-scale=1.0">
6      <title>Test helloworld</title>
7    </head>
8    <body>
9      <div id="app"></div>
10   </body>
11 </html>
12
```

src/main.js

main.js est le fichier "racine" de l'application, c'est à dire le point d'entrée du code JavaScript de toute l'application

```
JS main.js x
1 // The Vue build version to load with the `import` command
2 // (runtime-only or standalone) has been set in webpack.base.conf with an alias.
3 import Vue from 'vue'
4 import App from './App'
5 import router from './router'
6
7 Vue.config.productionTip = false
8
9 /* eslint-disable no-new */
10 new Vue({
11   el: '#app',
12   router,
13   components: { App },
14   template: '<App/>'
15 })
```

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

**Composants
Mono-Fichier**

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

CDN Vs Vue-CLI : Les composants mono-fichiers 🔍
=> <https://fr.vuejs.org/v2/guide/single-file-components.html>

```
▼ HelloWorld.vue x
1  <template>
2    <div>
3      <span>Hello {{who}}</span>
4    </div>
5  </template>
6
7  <script>
8    export default {
9      name: 'HelloWorld',
10     data () {
11       return {
12         who: 'World'
13       }
14     }
15   }
16 </script>
17
18 <style scoped>
19   span {
20     color: blue;
21   }
22 </style>
```

Un composant est un fichier
.vue composé de trois
éléments (optionnels)

<template> contient le code
HTML du composant

<script> contient le code
JavaScript du composant

<style> CSS du composant

La partie <script> du composant doit exporter un objet avec les propriétés du composant. On retrouve ici la propriété name, utile pour identifier le composant (appelé aussi selecteur) , et la propriété data qui fixe les données initiales du composant

App.vue

Composant racine de l'application dans lequel on peut donc mettre du code HTML (le template), du code JavaScript (les données, les méthodes, les événements. . .) et du code CSS

```
▼ App.vue x
1  <template>
2    <div id="app">
3      
4      <!--<router-view/>-->
5      <HelloWorld/>
6    </div>
7  </template>
8
9  <script>
10   import HelloWorld from './components/HelloWorld'
11   export default {
12     name: 'App',
13     components: {
14       HelloWorld
15     }
16   }
17 </script>
18
19 <style>
20 #app {
21   font-family: 'Avenir', Helvetica, Arial, sans-serif;
22   -webkit-font-smoothing: antialiased;
23   -moz-osx-font-smoothing: grayscale;
24   text-align: center;
25   color: #2c3e50;
26   margin-top: 60px;
27 }
28 </style>
29
```

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

**Composants
Mono-Fichier**

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Entrons un peu plus dans le détail 🔍

=> <https://fr.vuejs.org/v2/guide/components.html>

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

**Composants
Mono-Fichier**

Sujet 1

Les Filtres

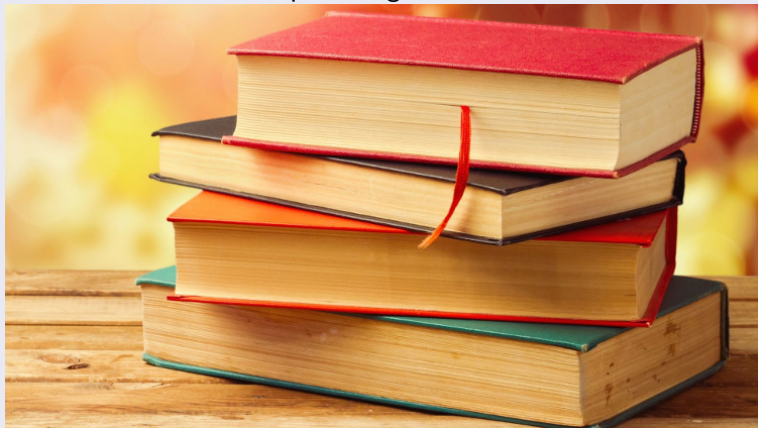
Le routing

Sujet 2

C'est pas fini

Exercice : Création d'une bibliothèque en ligne

Création d'une bibliothèque en ligne



Création d'une Bibliothèque en ligne : objectif

Détail du rayon : Nombre de livres likes = 2

Le Petit Prince - Antoine de Saint-Exupéry - 300 - 1943 liker!

Les fleurs du mal - Charles baudelaire - 635 - 1857 disliker!

Le rouge et le noir - Stendhal - 380 - 1831 disliker!

- Créer un projet, via CLI, nommé **Librairie**
- Créer un composant **Livre** : un livre est composé d'un titre, un Auteur, un nombre de pages, une année de parution et un critère like (booléen)
- Créer un composant **Rayon** : un rayon est composé de livre (liste statique initialisée dans le code)
- Par défaut, la page d'accueil correspond à la vue d'un rayon affichant les vues des Livre qui le composent (utiliser une directive pour le parcours de la liste) ainsi que le nombre total de livres likés. Le projet **Librairie** affiche uniquement cette vue. Chaque livre liké se voit affublé d'un coeur
- La vue d'un Livre doit proposer un bouton cliquable liker! ou disliker! selon l'état actuel du Livre

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

indice

- Une relation Parent-Enfant existe entre un Rayon et un Livre
- Les boutons Like / Dislike n'apparaissent que selon une condition

Création d'une Bibliothèque en ligne : corrigé

- vue init webpack librairie
- cd librairie
- npm install
- npm run dev (=> localhost : 8080) ...

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

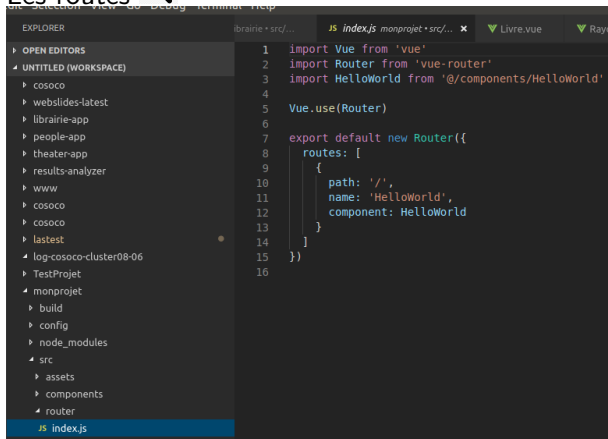
Sujet 2

C'est pas fini

Les filtres 🔍

=> <https://fr.vuejs.org/v2/guide/filters.html>

Les routes



The screenshot shows a VS Code editor interface. On the left, the 'EXPLORER' sidebar displays a file tree for a project named 'monprojet'. The tree includes folders like 'cosoco', 'websites-latest', 'librairie-app', 'people-app', 'theater-app', 'results-analyzer', 'www', and 'src'. The 'src' folder is expanded, showing subfolders 'assets', 'components', and 'router'. The 'router' folder is selected. In the center, the 'index.js' file is open, showing the following code:

```
1 import Vue from 'vue'
2 import Router from 'vue-router'
3 import HelloWorld from '@components/HelloWorld'
4
5 Vue.use(Router)
6
7 export default new Router({
8   routes: [
9     {
10       path: '/',
11       name: 'HelloWorld',
12       component: HelloWorld
13     }
14   ]
15 })
```

=>

<https://router.vuejs.org/fr/guide/essentials/named-routes.html>

Création d'une dvdthèque en ligne



| | | | | | | | |
|-------|--|-------|--|--------|--|---------|--|
| Titre | | Année | | Langue | | Acteurs | |
|-------|--|-------|--|--------|--|---------|--|

Liste des films

Juste la fin du monde (fr)

["Nathalie Baye", "Vincent Cassel"]

2016

Aperçu

Trois souvenirs de ma jeunesseE (fr)

["Quentin Dolmaire", "Lou Roy-Lecollinet", "Mathieu Amalric"]

2015

Aperçu

La La Land (en)

- Créer un projet, via CLI, nommé Dvdtheque
- Votre site est composée de 3 pages : Home et About proposant le contenu de votre choix, ainsi qu'une page Movies qui affiche une liste de films
- Nous n'utilisons pas de base de données : Les films sont issus d'un fichier json que vous aurez au préalable créer et enregistrer dans `assets/data/films.json`
- Toutes les pages du site doivent proposer une barre de navigation permettant de se rendre sur n'importe quelle autre page
- Pour chaque film, un lien Aperçu propose l'affiche du film en question sur une nouvelle page / composant spécifique (les affiches des films sont des images pouvant être stockées dans le répertoire `assets/images/` par exemple)

- La page movies propose des critères de recherche permettant de filtrer les films affichés, selon les titres, langues, acteurs et année de sortie (utiliser des filtres)
- Les première et dernière lettres des titres des films doivent être en majuscules (utiliser un filtre)
- Le survol d'un titre de film doit le colorer en vert (utiliser une directive)
- Liste des films doit être coloré en rouge (appliquer un css grâce à une directive)

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

indice

- Pour se rendre sur le composant affichant une image de films, penser aux routes avec indices
- Coupler les filtres à des valeurs calculées (computed)

Le framework
Vue.JS

Nicolas PARIS

Références

Introduction

Installation[1]

MVVM

HelloWorld

Les directives

Composants

Install[2] / CLI

Composants
Mono-Fichier

Sujet 1

Les Filtres

Le routing

Sujet 2

C'est pas fini

Formulaires et validation 🔍

=> <https://fr.vuejs.org/v2/cookbook/form-validation.html>

Date examen : 06/01/2021

=> <https://v3.vuejs.org/guide/introduction.html>

- Les grands concepts de Vue.js (SPA, organisation d'un projet, ...)
- Qu'est ce que le modèle MVVM ?
- Qu'est ce qu'un composant ?
- Les communications entre composants (parents et enfants)
- Qu'est ce qu'une directive ? (connaître les plus utiles : for, if, model, on, ...)
- Qu'est ce qu'un filtre ?
- Comment faire les routes ?
- ...