

Formation Javascript VueJs

Sommaire

- ▶ Rappel Javascript
- ▶ Rappel Javascript ES6+
- ▶ Introduction VueJs et Comparaison avec les autres frameWorks
- ▶ Utilisation de VueJs en bibliothèque
- ▶ Un premier Exemple
- ▶ Outils Debug et Dev VueJs
- ▶ Notion MVVM VueJs
- ▶ Directives VueJs
- ▶ Gestion des évènements VueJs
- ▶ Gestion des formulaires
- ▶ CLI VueJS

Sommaire

- ▶ Les Composants en VueJs
- ▶ Props des Composants
- ▶ Computed des Composants
- ▶ Communication entre Composants Parents et Enfants
- ▶ Création des événements
- ▶ Communication entre composants enfants et Parents
- ▶ Communication entre composants Frères

Sommaire

- ▶ Routage en VueJs
- ▶ Protection des routes en VueJs
- ▶ Cycles de Vie dans VueJS
- ▶ Communication Avec un serveur distants
- ▶ Utilisation des Mixins
- ▶ Vuex

Rappel Javascript

- ▶ Base de Javascript
- ▶ Javascript et POO
- ▶ Manipulation du DOM en Javascript

Rappel Javascript ES6+

- ▶ Introduction des let et const
- ▶ Destructeur
- ▶ Opérateur spread
- ▶ Opérateur rest
- ▶ Expression lambda
- ▶ Classes
- ▶ Import et export

Introduction VueJs et Comparaison Entre Framework

- ▶ Vue est un framework progressif pour la création d'interfaces utilisateur
- ▶ Vue se concentre sur la couche Vue et facilement intégrable avec des projets existants
- ▶ Vue est capable de créer des projets sophistiqués
- ▶ Vue est plus simple et flexible
- ▶ Vue vous permet de créer uniquement des parties spécifiques de votre application. Vous apprenez exactement ce qui est nécessaire pour le problème auquel vous faites face.
- ▶ Ou si c'est nécessaire et que vous avez le temps, vous pouvez en savoir plus et créer une application frontale complète et complexe à 100% Vue

Introduction VueJs et Comparaison Entre Framework



Vue.js



- ▶ Vue utilise Javascript, Angular s'appuie sur TypeScript
- ▶ Vue est assez facile à apprendre
- ▶ Vue est indiqué si vous travaillez seul ou si vous avez une petite équipe, Angular est pour de très grandes applications

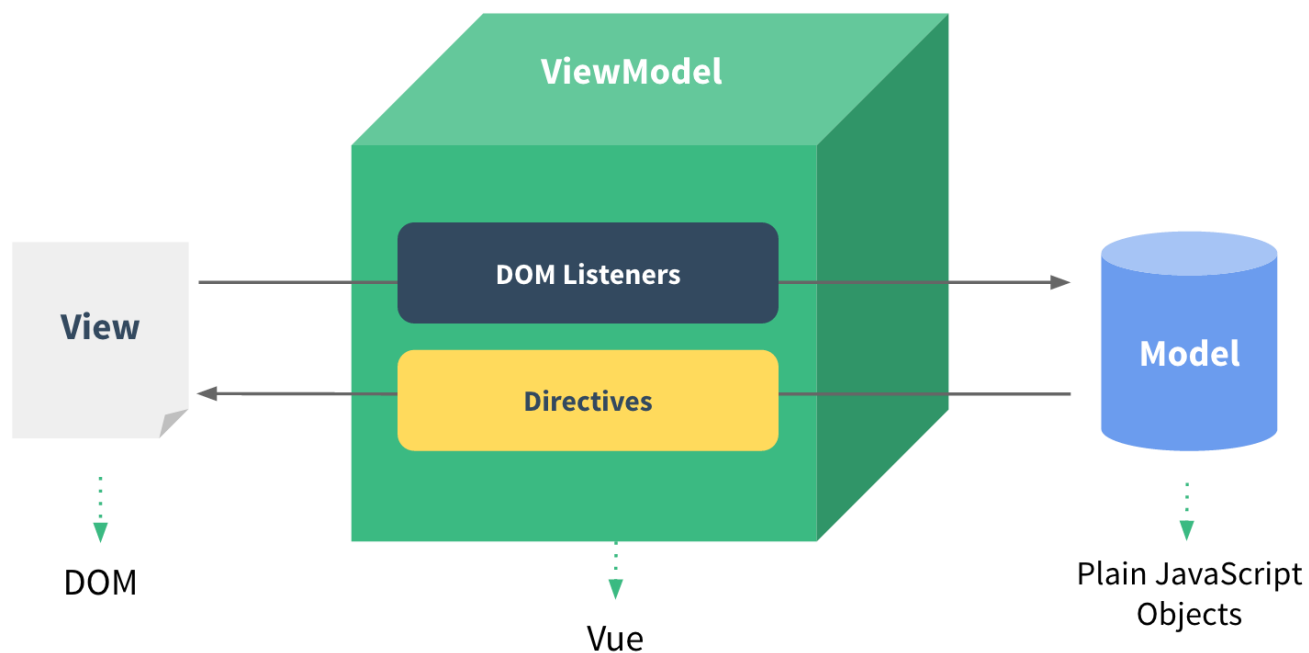
Un premier exemple

- ▶ Installation de VueJs en tant que bibliothèque
- ▶ Création d'un objet Vue
- ▶ Affichage des données dans le DOM

Outil de Debug et Dev VueJs

- ▶ Installation des extensions VueJs sur Vscode
- ▶ Installation des vuesdevTools sur les navigateurs

Notion Du MVVM Dans VueJs



Directives VueJs

- ▶ Directive conditionnelles
- ▶ V-if
- ▶ V-else
- ▶ V-else-if
- ▶ V-Show

Directive VueJs

- ▶ Directive boucle
- ▶ V-for

Directive VueJs

- ▶ V-model
- ▶ V-bind

Gestion des évènements

- ▶ VueJs permet d'écouter les évènements du DOM à l'aide des directives
- ▶ V-on
- ▶ Utilisation des modificateurs d'évènements

Exercices

- ▶ Exercices et TP

VueJS CLI

- ▶ VueJs CLI permet de développer des applications SPA avec des commandes
- ▶ VueJS CLI utilise node.js
- ▶ Node.js est une plateforme de développement javascript qui exécute du javascript sur n'importe quel environnement en utilisant google engine v8
- ▶ NPM est le gestionnaire de package de node.js
- ▶ VueJS n'est pas fait à l'aide de node.js, mais utilise le gestionnaire de package npm

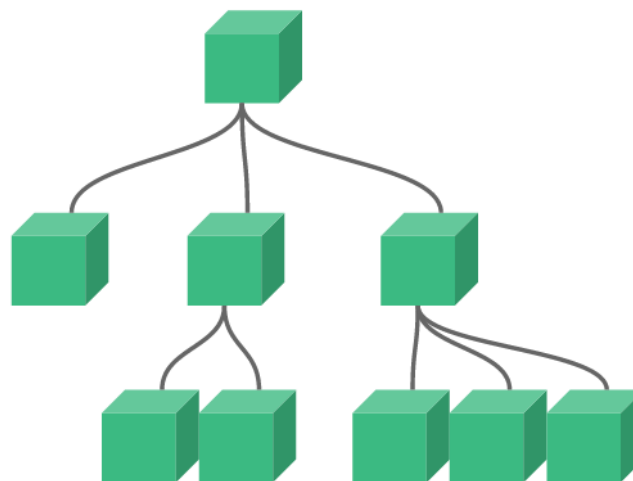
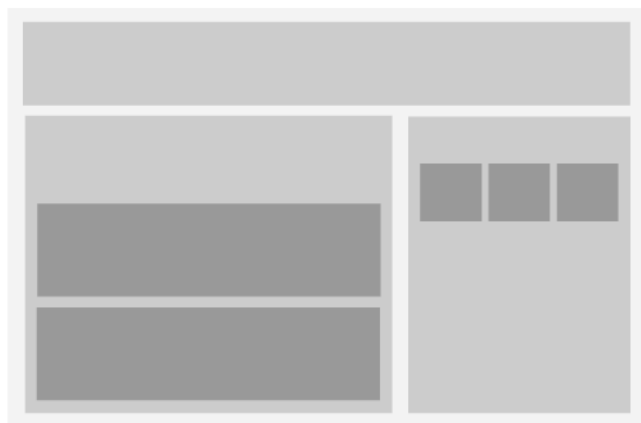
VueJS CLI

- ▶ Après installation de node.js on installe vuejs CLI
- ▶ `npm install -g vue-cli`
- ▶ Pour la création d'un nouveau projet en vuejs cli, on utilise la commande `vue init <nom-projet>`
- ▶ Pour démarrer un projet on utilise `npm run`

Composants en VueJs

- ▶ Les composants sont des instances de Vue réutilisables avec un nom.
- ▶ Les composants acceptent les mêmes options que Vue comme (data, methods...)
- ▶ Les seuls exceptions sont des options spécifiques à la racine comme el
- ▶ Comme les composants sont des instances réutilisables, l'option data doit être une fonction qui retourne un objet

Composant VueJs



Composant Vue JS

- ▶ Props:
- ▶ Les props sont des attributs personnalisables que vous pouvez enregistrer dans un composant
- ▶ Quand une valeur est passée à un attribut prop, elle devient une propriété de l'instance du composant
- ▶ Exemple d'utilisation des props statique au dynamique
- ▶ Les props forment **un flux de donnée descendant unidirectionnel**
- ▶ Les props sont utilisés pour passer des données d'un composant enfant vers un composant parent

Composants VueJS

- ▶ Propriété calculée
- ▶ Les propriétés calculées sont utilisées pour le rendu complexe d'un props
- ▶ Les propriétés calculées utilisent un système de mise en cache
- ▶ Définition des waths et comparaison avec les propriétés calculées

Composants VueJS

- ▶ Pour envoyer des données d'un composant enfant vers un composant parent, on utilise des évènements personnalisés.
- ▶ Pour créer un évènement, on utilise la méthode `$emit`
- ▶ On injecte dans l'évènement du composant enfant une méthode pour récupérer les données

Composants VueJS

- Pour faire communiquer deux composants frère, on utilise un objet de type Vue et les évènements emit et on

Routage VueJs

- ▶ Pour un créer un système de navigation entre les composants dans une application vueJs, on utilisera vue-router
- ▶ Vue-router peut être ajouter entant que librairie avec le lien
- ▶ `<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>`
- ▶ Vue-router peut être installer avec npm
- ▶ `npm install vue-router`
- ▶ Premier exemple

Routage VueJs

- ▶ Utilisation des paramètres dans url
- ▶ Utilisation des routes imbriquées
- ▶ Utilisation des intercepteurs de navigation pour protéger des urls
- ▶ Exemples

Cycle de Vie VueJS

- ▶ Une instance Vue passe par :
- ▶ `beforeCreate`
- ▶ `Created`
- ▶ `beforeMount`
- ▶ `Mounted`
- ▶ `beforeUpdate`
- ▶ `Update`
- ▶ `beforeDestroy`
- ▶ `Destroyed`

Communication avec application distante

- ▶ En utilisant fetch
- ▶ En utilisant une bibliothèque externe comme axios
- ▶ Exemple d'utilisation de la librairie axios

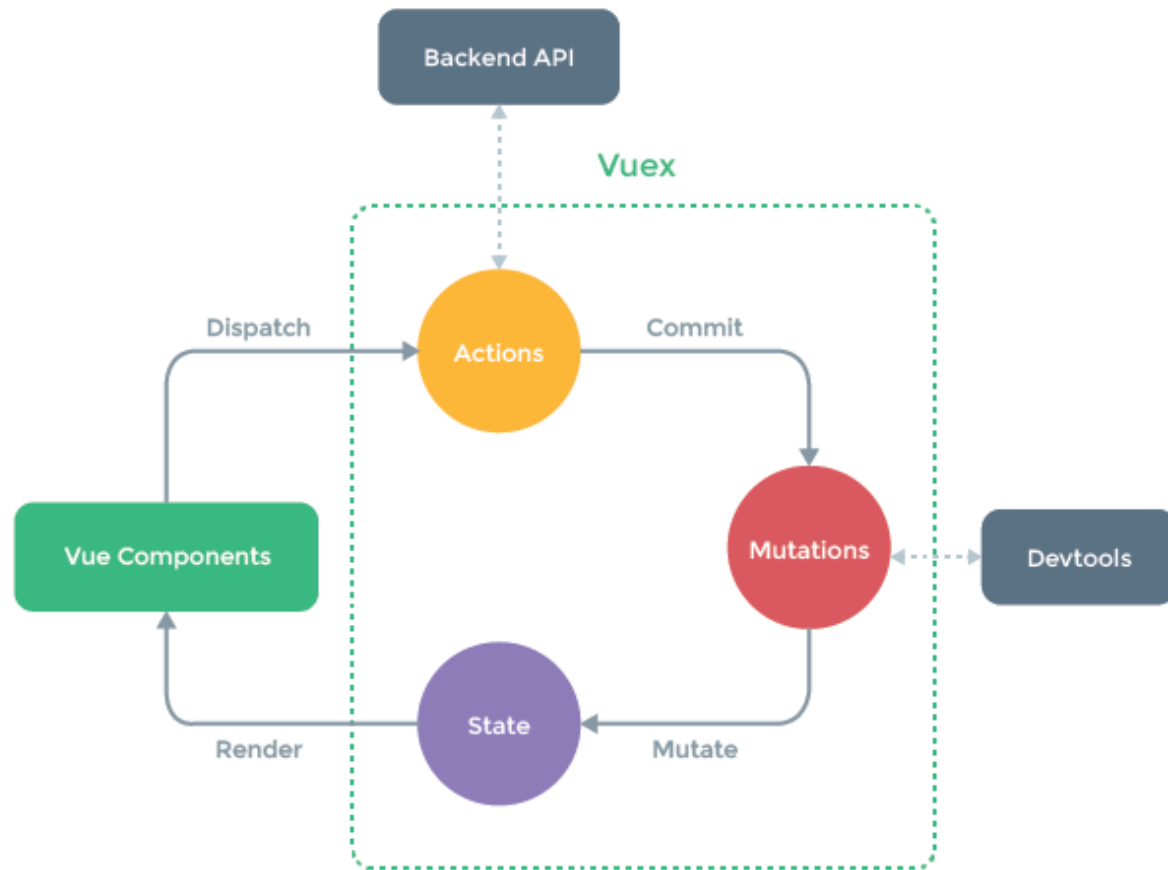
Mixins

- ▶ Les mixins offrent la possibilité de créer des fonctionnalités réutilisables pour les composants de vue
- ▶ Un objet mixin peut contenir toute option valide pour un composant
- ▶ Quand un composant utilise un mixin, toutes les options du mixin seront “fusionnées” avec les options du composant
- ▶ Exemple d'utilisation des mixins

VueX

- ▶ Vuex est un **gestionnaire d'état** et une **bibliothèque** pour des applications Vue.js
- ▶ Il sert de zone de stockage de données centralisée pour tous les composants dans une application
- ▶ Il utilise des règles pour s'assurer que l'état ne puisse subir de mutations que d'une manière prévisible

VueX



VueX

- ▶ Store
- ▶ State
- ▶ Mutations
- ▶ Actions