VUE

Histoire du framework

Vue a été créée par Evan You après avoir travaillé pour Google en utilisant AngularJS dans un certain nombre de projets. Il a ensuite résumé son processus de réflexion : Je me suis dit : "Et si je pouvais juste extraire la partie que j'aime vraiment dans Angular et construire quelque chose de vraiment léger". Le premier commit de code source du projet était daté de juillet 2013, et Vue a été publié pour la première fois en février suivant, en 2014.

Plaisir au travail

70% des personnes utilisant Vue JS prennent du plaisir à utiliser ce framework, sont satisfaits, d'après les statistiques de Stackoverflow sur les 3 dernières années.

Popularité

Vue étant le plus jeune des trois frameworks, il est un peu en retrait sur les recherches Google, et tourne autour de 25% de popularité d'après les statistiques de Google Trends sur les 12 derniers mois . Par contre, on voit qu'il est téléchargé en moyenne 1 000 000 de fois par mois (sur les deux dernières années) via le gestionnaire de paquet npm , et se situe donc au même niveau qu' Angular.

Emploi

Vue n'est pas le framework le plus recherché sur le marché de l'emploi, devancé de loin par React et Angular .
Vue fait souvent un peu plus peur aux entreprises car il n’est pas soutenu par un GAFAM. Il faut bien comprendre que l’adoption d’une technologie pour une entreprise en place prend énormément de temps (plusieurs années souvent).

Facilité, temps d'apprentissage

Contrairement à React et Angular, Vue semble plus facile à prendre en main. Les grandes lignes de l'apprentissage du framework peuvent se décomposer comme suit :

- Premières directives
- Compréhension du système de composant
- Routing, communication entre les composants
- Optimisation

Pour ce qui est de Vue.js, c’est simple au début, au milieu et à la fin. Il n’y aura pas vraiment de point de blocage, de moments de souffrance intense en ayant l’impression de ne rien comprendre. Il y a quand même toutes les fonctionnalités de base à apprendre mais c’est vraiment accessible et le framework est beaucoup moins exigeant sur votre niveau en JavaScript que React. Vue.js a clairement un avantage ici, et cela explique pourquoi son adoption par les débutants est très importante et que le framework a réussi à se faire une place. Surtout lorsque l’on sait que cette simplicité n’impacte pas pour autant ses performances.

Structure d'un projet type

Le projet type se compose :

- d'un dossier node_module, où le gestionnaire de package s'occupe de tout, et installe les dépendances, et modules.
- d'un dossier public
- d'un dossier src : dans lequel on retrouve la principale architecture d'un site :
- assets
- components
- app.vue
- main.js

On y ajoutera souvent le dossier "views", "router"

Ensuite on trouve le fichier package.json, dans lequel sont listées toutes les dépendances, avec leur versions installées sur le projet.

- un README (optionnel)
- des fichiers de configuration.

Langages, dépendance

Seulement Javascript est nécessaire pour utiliser le framework, mais Typescript peut également être utilisé.

Pour les dépendances, voici à quoi ressemble le package.json d'un projet vue3, créé par Vue-CLI, en choisissant "par défaut" :


```
        "dependencies": {
        "core-js": "^3.6.5",
        "vue": "^3.0.0"
 },
 "devDependencies": {
        "@vue/cli-plugin-babel": "~4.5.0",
        "@vue/cli-plugin-eslint": "~4.5.0",
        "@vue/cli-service": "~4.5.0",
        "@vue/compiler-sfc": "^3.0.0",
        "babel-eslint": "^10.1.0",
        "eslint": "^6.7.2",
        "eslint-plugin-vue": "^7.0.0"
 },
 "eslintConfig": {
        "root": true,
        "env": {
        "node": true
        },
        "extends": [
        "plugin:vue/vue 3-essential",
        "eslint:recommended"
        ],
        "parse Options": {
        "parser": "babel-eslint"
        },
        "rules": {}
 },
 "browserslist": [
        "> 1%",
        "last 2 versions",
        "not dead"
 ]
```

Manière de créer un composant / Nature du composant

Sur Vue, l'intégralité d'un composant se trouve sur une seule page.
Il peut se décomposer en trois parties :

- une partie template : dans lequel se trouve du html. On relie cette template aux méthodes, propriétés et données présentes dans la partie script via des interpolations. Vue dispose aussi de nombreuses fonctions accessibles directement depuis ce template.

- une partie script : où l'on importe/export les composants, déclare les méthodes, les données, les propriétés du composant. On peut également importer des modules entiers, les paramétrer dans ces balises.

- une partie style : écrite en css, scss, sass ou encore sass. Définit le style des composants.

Pour créer le composant il suffit de créer un fichier .vue, ensuite le composant est réutilisable et instanciable dans n'importe quelle vue.

REACT

Histoire du framework

React est sans doute, au regard du nombre de ses téléchargements, la bibliothèque JavaScript la plus populaire.
Basé sur un système de composants et permettant de faciliter la création d'éléments voués à l'interface utilisateur (UI), il est particulièrement recommandé pour le développement d'applications web et mobiles.
Il a également l'avantage de bénéficier du support d'une entreprise dont la connaissance d'Internet n'est plus à démontrer puisqu'il est développé et maintenu par Facebook

Naissance de React

React, parfois également appelé
React.jsouReactJS, a été développé en 2013 par Jordan Walke.
Il est alors ingénieur chez Facebook et travaille sur Facebook Ads. L'équipe du réseau social comprend rapidement l'intérêt de cet outil interne, et souhaite l'utiliser pour sa propre application. Il est depuis lors développé et maintenu par Facebook.
Le but de Jordan Walke est de structurer des applications en JavaScript. Il décrit tout d'abord React comme une bibliothèque JS conçue pour développer des interfaces utilisateurs.
Initialement publié sous licence BSD modifiée en 2014, React est depuis 2017 distribué sous licence MIT.
Cela fait de lui une bibliothèque libre et open source pleinement utilisable, y compris à des fins commerciales.
Grâce au pouvoir d'action d'une multinationale comme Facebook,React est très régulièrement revu, corrigé et complété de nouvelles fonctionnalités.
En contrepartie, les applications web développées à l'aide de React doivent régulièrement être mises à jour, ce qui ne pose cependant pas de problème technique majeur.

Plaisir au travail

C'est actuellement le framework le plus utilisé en France et dans le monde : Facebook, Instagram, Airbnb, Netflix, Paypal, Uber, le New York Times, WhatsApp, Leboncoin et de nombreuses autres entreprises l'ont adopté. Côté performance, le framework est très léger et extrêmement rapide.

Ce qui distingue React
L'idée sous-tendant la création de React est de parvenir à améliorer les performances des applications web en limitant l'accès au DOM(pour Document Object Model), qui dé□nit la structure de la page. React se base donc sur une approche par composant permettant de ne charger que les éléments changeants et non la page complète à chaque interaction.
React n'est donc pas un framework mais une bibliothèque utilisée pour coder et af□cher des composants. C'est l'une de ses particularités face à ses principaux compétiteurs comme Angular et Vue. React ne dispose pas d'autres fonctionnalités et se distingue par la même d'un framework qui impose un cadre de travail complet en fournissant les outils qui vont avec. Il permet un développement front-end □uide et harmonieux, ce que les anglophones nomment seamless, littéralement« sans couture ».
L'expérience utilisateur(UX) passe avant tout La force de React repose donc sur ses composants, des éléments réutilisables de l'UI. Par défaut, le HTML dispose de certains composants, toutefois basiques et limités en nombre. React vient combler ce manque en permettant le développement de composants additionnels en JavaScript. En tant que développeur, cela permet, en n'utilisant que du JavaScript, de créer des éléments de page

qui vont collaborer, voire générer des éléments en HTML et CSS En addition, React introduit le JSX, un langage spécifique simplifiant la syntaxe du JavaScript pour la rapprocher du HTML. Grâce à JSX, le code produit est plus lisible et facile à maintenir.
React est donc un outil complexe, qui peut toutefois être utilisé de façon simple. JSX est par exemple optionnel. Tout ne doit pas non plus être créé à partir de React qui peut facilement s'intégrer à certaines parties d'une application ou tout simplement être ajouté à une application existante

Popularité

React au service des entreprises
React est très populaire et plébiscité par de nombreuses entreprises,quelle que soit leur taille. Cela s'explique en partie par le fait qu'il soit centré UI/ UX et permette donc le développement d'applications qui plaisent à leurs utilisateurs. Cela implique tout d'abord un marché de l'emploi plutôt plus riche que les autres frameworks JavaScript. Cela se traduit également par l'existence d'une documentation détaillée, servi par une
large communauté, et bien sûr par Facebook. La présence d'une entreprise de cette taille derrière la bibliothèque garantit la stabilité et la pérennité de la technologie
, un avantage non négligeable tant pour les développeurs que les entreprises l'utilisant.

Emploi Facilité,

Connaître les compétences générales des développeurs Web telles que HTML, CSS et JavaScript est indispensable pour travailler en tant que développeur Web, mais plus vous ajoutez de compétences telles que React à vos fondations, plus vous serez demandé par les employeurs et plus vous pouvez gagner d'argent. Et quand vous regardez le temps qu'il faut pour acquérir une compétence comme React JS (littéralement PLUSIEURS mois), cet investissement devient une évidence. Être un développeur Web à succès consiste en partie à rester à la pointe des technologies actuelles, et React est actuellement au sommet de la technologie Web.

Pour mettre les choses en perspective, une recherche d'emplois de développeur React sur Indeed.com à ce jour fait apparaître près de 14 000 offres d'emploi ouvertes.
(s'ouvre dans un nouvel onglet) dont le salaire varie de 85 000 $ à 130 000 $ par an. Même au bas de l'échelle, c'est 10 000 $ de plus que l'estimation moyenne actuelle d'Indeed pour les salaires des développeurs Web généraux (73 000 $ par année).

Temps d'apprentissage

React propose un guide de démarrage qui devrait vous aider à configurer React en une heure environ. La documentation est complète et complète, avec des solutions aux problèmes courants déjà présents sur Stack Overflow.
React n'est pas un framework complet et les fonctionnalités avancées nécessitent l'utilisation de bibliothèques tierces. Cela rend la courbe d'apprentissage du framework de

base moins raide, mais dépend du chemin que vous empruntez avec des fonctionnalités supplémentaires. Cependant, apprendre à utiliser React ne signifie pas nécessairement que vous utilisez les meilleures pratiques.

React est juste assez vieux pour être mature et a un grand nombre de contributions de la communauté. Il a été largement accepté. Le marché du travail pour React est vraiment bon et l'avenir de ce framework s'annonce prometteur.

React semble être un bon choix pour quelqu'un qui s'initie aux frameworks JavaScript frontaux, aux startups et aux développeurs qui aiment une certaine flexibilité. La possibilité de s'intégrer de manière transparente avec d'autres frameworks lui confère un grand avantage pour ceux qui souhaitent une certaine flexibilité dans leur code.

Structure d'un projet type

Grouper par fonctionnalité ou par route
Une façon courante de structurer les projets consiste à placer le CSS, le JS et les tests ensemble dans des dossiers groupés par fonctionnalité ou par route.

```
common/
  Avatar.js
  Avatar.css
  APIUtils.js
  APIUtils.test.js
feed/
  index.js
  Feed.js
  Feed.css
  FeedStory.js
  FeedStory.test.js
  FeedAPI.js
profile/
  index.js
  Profile.js
  ProfileHeader.js
  ProfileHeader.css
  ProfileAPI.js
```

La définition d’une « fonctionnalité » n’est pas universelle, et il vous appartient d’en définir la granularité. Si vous ne parvenez pas à créer une liste des dossiers racines, vous pouvez demander aux utilisateurs de votre produit quelles en sont les principales composantes et utiliser leur modèle mental comme base.

Grouper par type de fichier

Une autre manière répandue de structurer les projets consiste à grouper les fichiers similaires ensemble, par exemple :

```
api/
```

```
  APIUtils.js
  APIUtils.test.js
  ProfileAPI.js
  UserAPI.js
components/
  Avatar.js
  Avatar.css
  Feed.js
  Feed.css
  FeedStory.js
  FeedStory.test.js
  Profile.js
  ProfileHeader.js
  ProfileHeader.css
```

Certaines personnes préfèrent également aller plus loin et séparer les composants dans des dossiers différents en fonction de leur rôle dans l'application. Par exemple, la conception atomique est une méthode de conception reposant sur ce principe. N'oubliez pas qu'il est souvent plus productif de traiter de telles méthodologies comme des exemples utiles plutôt que comme des règles strictes à suivre.

Évitez trop d'imbrication

L'imbrication excessive de répertoires dans les projets JavaScript est source de nombreuses souffrances. Il devient plus difficile d'écrire des importations relatives entre eux ou de mettre à jour ces importations lorsque les fichiers sont déplacés. À moins d'avoir une excellente raison d'utiliser une structure de dossiers profonde, limitez-vous à un maximum de trois ou quatre imbrications de dossier dans un même projet. Bien entendu, il ne s'agit que d'une recommandation, qui n'est peut-être pas pertinente pour votre projet.

Si vous démarrez tout juste un projet, ne passez pas plus de cinq minutes à choisir une structure de fichiers. Choisissez l'une des approches ci-dessus (ou prenez la vôtre) et commencez à écrire votre code ! Vous serez sûrement amené·e à la repenser de toutes façons une fois que vous aurez produit du vrai code.
Si vous vous sentez complètement bloqué·e, commencez par garder tous les fichiers dans un seul dossier. Si votre application se met à grossir, vous aurez envie de séparer certains fichiers du reste. À ce moment-là, vous aurez une bonne idée des fichiers que vous éditez ensemble régulièrement. En général, il est judicieux de conserver en un même endroit des fichiers qui changent souvent ensemble. Ce principe s'appelle « colocation ».
À mesure que les projets prennent de l'ampleur, on utilise souvent un mélange des deux approches ci-dessus. Du coup, choisir la « bonne » au début n'est pas si important.

Langages, dépendance

Manière de créer un composant

Étape 1 - Mettre en place le projet React
Étape 2 - Créer un composant indépendant avec des classes React
Étape 3 - Créer une structure de fichier lisible
Étape 4 - Construire un composant fonctionnel

Vous avez maintenant une petite application avec des pièces indépendantes. Vous avez créé deux grands types de composants : fonctionnels et de classe. Vous avez séparé des parties des composants dans des répertoires afin de pouvoir conserver des morceaux de code similaires regroupés. Vous avez également importé et réutilisé les composants.
En comprenant les composants, vous pouvez commencer à considérer vos applications comme des pièces que vous pouvez démonter et remonter. Les projets deviennent modulaires et interchangeables. La capacité de voir des applications entières comme une série de composants est une étape importante dans pour penser dans un environnement React

Les dépendances de production :

Jests est généralement inclus dans un projet create-react-app. C'est une librairie permettant de tester nos composants et notre application

React

La librairie en elle même avec toutes les fonctionnalités et avantages qu'elle contient

React-dom

React-Dom permet d'utiliser les fonctionnalités de la librairie sur le navigateur contrairement à React-Native qui permet de les utiliser sur smartphone.

ANGULAR

Histoire du framework

Qu’est-ce que Angular ?

Angular est une plateforme (ou framework) Javascript pour créer des interfaces utilisateurs. Il permet aussi de créer des applications natives pour les ordinateurs de bureau et les téléphone

Brève histoire du web

Vers le début des années 2000 Javascript n'était pas aussi avancé qu’il ne l’est maintenant et il était nécessaire mais très difficile de coder dans ce langage pour manipuler le DOM et faire que les sites fonctionnent de la même façon sur tous les navigateurs.

Heureusement, des développeurs se sont réunis pour créer une librairie qui pourrait fonctionner sur tous les navigateurs : Jquery.

À cette époque, Jquery était très important pour tous les sites pour s’assurer qu’il fonctionne comme prévu et sur tous les navigateurs. Certains développeurs n'apprennaient même pas le javascript en lui même mais Jquery directement.

Javascript ayant pris de l'ampleur, il a été amélioré pour ses concepteurs pour faciliter la manipulation du DOM. Les navigateurs se sont aussi améliorés en parallèle pour se standardiser.

Aujourd’hui la mort de Jquery est prévue par beaucoup au profit des frameworks comme Angular.

Les frameworks et librairies

Les frameworks et librairies sont des collections de code pré-écrit pour augmenter le “niveau d'abstraction" du code.

Les frameworks quant à eux donnent une frame (cadre) ou une façon d’écrire du code bien précise et bien organisée. Ils sont conçus par des développeurs croyant que les applications doivent fonctionner et être écrites d’une façon bien précise.

Les débuts d’Angularjs

Angular a été fondé par Misko Hevery et Adam Abrons qui ont baptisé le framework angularjs, car il fonctionne avec HTML qui a des balises avec des "angle brackets" ou des crochets en angle.

Il est apparu en 2010 en même temps que des frameworks comme backbone ou ember mais avec des particularités. Mais quelles étaient-elles ?

Liaison des données en deux parties, les injections de dépendances et les directives.

Comme je l’ai dit plus haut, les frameworks définissent une manière bien spécifique de créer et d’organiser des applications.

- Liaison des données en deux parties

Dans le cas d’angularjs, il utilisait avant l’architecture MVC (modèle, vue, controlleur) emprunté au développement backend qui sépare les applications en ces trois parties distinctes.

Le modèle est la partie responsable des règles et de la logique de l'application.
La vue est simplement l’interface utilisateur.
Le contrôleur qui est la partie acceptant les inputs pour les transmettre au modèle.

La liaison en deux partie signifie que le modèle et la vue sont synchronisés

- Les injections de dépendances

Les dépendances sont des relations entre des différentes parties de code.
Traditionnellement, chaque objet dans du code doit créer un objet dépendant de lui.
Ce qui faisait que si on devait changer un objet, il fallait aussi changer sa dépendance

Angularjs utilisait une approche différente en utilisant des injecteurs qui lient les objets aux dépendances qui sont toutes stockées dans un même endroit. Cela permet de réutiliser du code plus facilement et de mieux le tester.

- Les directives

Les directives sont très utilisées dans Angular ils fonctionnent et s’écrivent comme des balises HTML mais précisent à quelles fonctionnalités doivent être attachées à un élément particulier. Ils permettent en autre de générer du contenu dynamique dans du code HTML.

Tout cela permettait de créer des applications en une seule page avec beaucoup d’éléments interactifs et résolvait beaucoup de problèmes que les ingénieurs rencontraient.

Toutefois les applications très larges étaient généralement lentes et Angularjs ne donnait aucune solution pour régler ce problème.

Mais, une autre technologie avait une solution.

Naissance de React

Réact a été créé par facebook en 2013 et est un autre librairie javascript. Elle offrait une meilleure solution que Angularjs pour régler les problèmes de conception, d'organisation et de rapidité des applications.

La librairie utilise la liaison de données en une partie, qui rassemble le modèle, la vue et le contrôleur et utilise des composants réglant les problèmes de rapidité et d'architecture des applications larges.

La même équipe a aussi créé react native qui ramène la puissance et les fonctionnalités de React sur les téléphones mobiles.

La renaissance d'Angularjs en Angular 2 ou simplement Angular

Face à la popularité et à la puissance de React l'équipe de google décida de complètement réécrire la plateforme

Le nouveau modèle d'Angular n'utilise plus l'architecture MVC, mais est aussi basée sur les composants tout comme React permettant de modifier une partie de la page sans la rafraichir

Elle utilise aussi 4 façons de lier les données permettant de mieux définir la communication entre les composants

Le framework utilise aussi Typescript qui est une version améliorée de Javascript
Cela a aussi permis aux développeurs de transitionner plus facilement vers Nativescript qui est un framework utilisant typescript permettant de créer des applications mobiles.

NGRX est un outil permettant de gérer les états (ou données) dans les composants de façon à les stocker dans un seul endroit au lieu de les faire circuler d'un composant à un autre.

L'injection de dépendances a aussi été améliorée pour booster la rapidité des applications. Maintenant les injecteurs sont hiérarchisés en arborescence permettant d'appliquer des dépendances et services à des groupes de composants.

Enfin, un service optionnel appelé Angular universal permet de faire du rendu côté serveur permettant d'améliorer la vitesse du rendu côté client et le SEO.

Et bien d'autres fonctionnalités comme le Lazy loading, la validation de formulaire , des animations, testing etc..

Plaisir au travail

Seulement 45% des développeurs disent aimer travailler avec angular

Popularité

Environs 243 000 posts sur stack overflow
Sur google trends on peut remarquer qu'Angular et React sont au coude à coude dans les résultats de recherche avec 70 % de recherche. 71 pour react.
Sur github il n'y a environ que 57000 étoiles pour Angular. 156000 pour vue js

Emploi Facilité,

En France en recherchant sur Indeed on trouve environs 2727 emplois pour angular

Temps d'apprentissage

Ce framework est le plus diffcile à apprendre, car il demande de connaître html, css, javascript et typescript
En plus la quantité de fonctionnalités qu'il contient fait qu'il est plus difficile de tout connaitre.

Structure d'un projet type

On peut y voir le fichier node_modules qui va contenir des programmes utilitaires pour développer et faire fonctionner notre application.

Le fichier src qui va contenir notre projet, nos composants, nos images, etc.

Le fichier .browserslistrc qui contient des informations permettant à angular de supporter différents navigateurs

Le fichier .editorconfig permettant de configurer notre environnement de travail

Le fichier .gitignore qui spécifier à Git ce qu'il ne faut pas enregistrer dans notre dépot

Le fichier angular.json qui fournit à Angular des configurations par défaut pour notre environnement de développement et pour le déploiement.

Le fichier karma.conf.js contient des configurations pour tester nos composants.

Les fichiers package-lock.json et package.json qui contient des scripts et les dépendances pour notre projet.

Des fichiers tsconfig permettant de configurer typescript

Langages, dépendance

Les languages utilisés sont HTML,CSS, Javascript et typescript

- Dépendances de production

angular/animations
angular/common
    angular/compiler
angular/core,
angular/forms
angular/platform-browser
angular/platform-browser-dynamic
angular/router
rxjs
tslib
zone.js

- Dépendances de développement

angular-devkit/build-angular
angular/cli
    angular/compiler-cli
types/jasmine
types/node
    jasmine-core
karma
    karma-chrome-launcher
karma-coverage
    karma-jasmine
    karma-jasmine-html-reporter
typescript

Manière de créer un composant

Pour générer un composant on peut rentrer la ligne :
ng generate component nom_de_dossier/nom_de_composant

- **La structure d'un composant**

Un composant est généralement composé de 4 fichiers :

- nom_du_composant.component.html
- nom_du_composant.component.css
- nom_du_composant.component.spec.ts
- nom_du_composant.component.ts

Le fichier html peut contenir :
Les balises html, les propriétés et les fonctions de notre composant

Le fichier css peut contenir :
Les propriétés css de notre composant

Le fichier specs.ts peut contenir :
Les configurations de test de notre composant, spécifiant ce que le composant doit faire et s'il réussit à le faire.

Le fichier component.ts peut contenir :
Les fonctions, les propriétés, les méthodes de cycle de vie, le nom de notre composant, etc.