

Documentation technique



Besjan Sejrani
besjan.sejrani@cpnv.ch

SI-CA1a
01.04.2020

Table des matières

Références.....	4
Résumé.....	5
Choix.....	5
Image format.....	6
Webp.....	6
Avantages	6
Inconvénients	6
Npm && Nodejs.....	7
Avantages	7
Inconvénients	8
Npm scripts.....	8
.....	9
Electron	9
Avantages	9
Inconvénients	9
Electron-builder	10
Github.....	10
.....	11
Sass	11
Avantages	11
Inconvénients	11
.....	12
Webpack	12
Avantages	12
Inconvénients	14
Threejs.....	15
Avantages	15
Inconvénients	15
Communication	16
Architecture MVC(Model, View, Controller)	16
API Rest (Application Programming Interface)	16
GraphQL	17
gRPC	18

Recommandations	19
Tests	19
Typescript	20
React	21
CI / CD	22
Heroku (Plateform as a Service)	23
Netlify	24

Références

Enfants du Monde	
Prénom et nom	Susana Borda Carulla
Email	susana.borda@edm.ch

Maître de classe	
Prénom et nom	Jean-Philippe Chavey
Email	jean-philippe.chavey@cpnv.ch

Elève	
Prénom et nom	Besjan Sejrani
Email	besjan.sejrani@cpnv.ch

Résumé

Choix

Tous les choix techniques effectués ont été fait dans l'unique but de pouvoir améliorer l'application pour qu'elle ait une grande portabilité à la fois sur les navigateurs et les systèmes d'exploitation, tout en suivant le cahier des charges.

La rapidité de développement ainsi que la restructuration du projet ont été effectué afin d'offrir l'ajout de fonctionnalités / mis à jour plus régulièrement.

Image format

Webp

Webp est un codec de Google développé en 2010, il permet notamment de compresser une image avec une efficacité d'environ 30% supérieur à celle du jpeg ou du png.

Dans le transfert de ressources via la bande passante, ce qui coûte le plus cher ce sont les images.

Etablir un moyen de diminuer la bande passante, permet à l'utilisateur de télécharger les ressources plus rapidement ainsi que d'améliorer l'expérience utilisateur afin qu'il continue d'utiliser l'application ou le site web.

De plus, diminuer la bande passante permettrait aux utilisateurs ayant un forfait limité en GB / mois d'économiser sur leur abonnement.

Avantages

- Compression efficace.
- Gain de temps de chargement.

Inconvénients

- En 2020, tous les navigateurs l'ont accepté, sauf Safari d'Apple. Par conséquent si quelqu'un avec le navigateur Safari se dirige sur le site web, il ne verrait rien.
- L'application du globe virtuel sur USB ne craint pas ce cas de figure, il est contenu dans Chromium, version open source de chrome.
- Une grande quantité de requêtes HTTP influencent également le temps de chargement d'une page, même si chaque ressource fait 1KB.



Npm & Nodejs

Node.js est un environnement de développement utilisant du Javascript côté serveur, qui jusqu'ici était limité aux navigateurs.

Grâce au moteur Javascript de Chrome, mis en open source et nommé V8, on peut créer des lignes de commandes, des serveurs web, des applications desktop et plein encore.

Lorsque l'on installe Nodejs, un package manager nommé NPM est par défaut installé, permettant d'installer via la ligne de commande n'importe quelle librairie / framework ou fichier Javascript mis publiquement sur le repo en ligne.

NPM est l'équivalent de Composer pour PHP, Gem pour Ruby, Pip pour Python, NuGet pour C#, ...

Lorsque l'on installe une librairie avec NPM, automatiquement une ligne s'ajoute à notre fichier packages.json avec le nom et la version de la librairie. Pour collaborer, on a juste à partager notre fichier JSON avec nos collègues pour qu'ils puissent intégrer les mêmes dépendances que l'on a via la commande `npm install`.

Il faut au préalable effectuer la commande `npm init` afin de générer le fichier JSON.

Avantages

- La communauté javascript est la première ou deuxième plus grande au monde, il existe une multitude de librairies permettant d'intégrer à notre code base afin de créer des projets de qualités.

Inconvénients

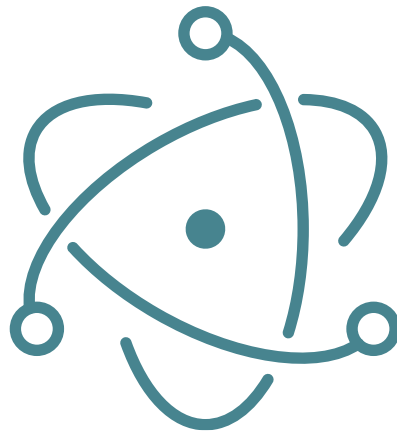
- Nodejs est très performant, cependant, il demande une courbe d'apprentissage non négligeable.

Npm scripts

Puisque Node.js est un environnement de développement côté serveur, il a accès à notre système de fichiers, et peut donc le modifier.

Un script NPM est l'équivalent d'un script Bash sous Linux ou PowerShell sous Windows.

Via les scripts, on peut faire faire à notre système ce que l'on désire.



Electron

Electron est un framework javascript permettant de créer des applications desktop pour les plateformes Windows, Linux et Mac Os, à partir d'une seule code base.

Développé par Github, electron est un mélange de Chromium (version open source de Google Chrome) et V8 (moteur Javascript open source de Google Chrome). Ce dernier possède toutes les API pour communiquer avec notre système d'exploitation de manière native.

Electron a notamment été utilisé pour créer Slack, Discord, Wordpress desktop, Postman, Visual Studio Code, Atom, ...

Avantages

- Développer et maintenir une application en electron est très simple, pour cela il suffit de connaître du HTML, CSS et un peu de Javascript.
- Gain de productivité et financier.
- Installer une application nativement permet d'éviter les erreurs de version de dépendances. C'est-à-dire, avoir deux versions PHP sur son ordinateur peut tout simplement provoquer des erreurs lors de l'exécution de certains programmes. Ces deux versions ne sont pas isolées l'une de l'autre.

Inconvénients

- Déployer une application en electron sous Microsoft Store et l'Apple Store est recommandé pour qu'un maximum d'utilisateurs téléchargent l'application, mais obtenir les certificats signés de ces autorités demandera du temps et de l'argent.
- Télécharger une application sans certificats d'autorités, provoquera l'apparition d'un pop-up afin d'avertir que notre application est potentiellement dangereuse pour l'utilisateur. Mécanisme de défense.

- Les systèmes d'exploitation a minima compatibles, sont Windows 7, Mac Os 10.10 (Yosemite) et Ubuntu 12.04.

Electron-builder

A l'aide d'electron-builder, simple librairie pour electron, on peut facilement mettre en avant à l'utilisateur que des nouvelles versions de l'application sont disponibles s'il a accès à Internet. S'il n'a pas accès à Internet, electron détectera automatiquement le moment où il en aura, et lui proposera à cet instant-là.

Pour cela, on effectue les changements que l'on désire, et on lance la commande prédéfinie à l'aide du script NPM dans le package.json, notre code est ensuite compilé pour les différentes plateformes et sera envoyé sur Github qui lui s'en chargera de distribuer et de notifier l'utilisateur qu'il y a eu un changement afin qu'il télécharge la nouvelle.

Github

Le rôle de Github dans electron est de substituer un update server ainsi que l'Apple Store et Microsoft Store.



Sass

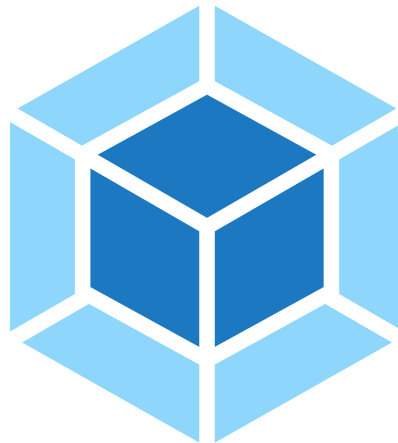
Sass est un préprocesseur CSS permettant d'ajouter des fonctionnalités productives en phase de développement, une fois compilé, Sass devient du CSS normal.

Avantages

- Permet d'utiliser des variables pour désigner du CSS
- Permet d'importer des fichiers SCSS entre eux
- Permet de mieux structurer les projets content beaucoup de CSS
- Permet d'utiliser des fonctions avec Sass
- Permet de compiler uniquement un seul fichier css, évitant des requêtes HTTP inutiles.

Inconvénients

- Légère courbe d'apprentissage



Webpack

Webpack est un « file bundler » permettant de combiner les fichiers Javascript et CSS ensemble, diminuant ainsi drastiquement les requêtes envoyées au serveur.

Mais Webpack ce n'est pas que cela uniquement. C'est avant tout un build tool ainsi qu'un écosystème.

Webpack utilise le principe de plugin et de loader, c'est-à-dire qu'au moment de compiler le projet, on peut lui dire ce que l'on aimerait qu'il fasse pour nous. La communauté javascript dépendant fortement de cet outil, elle a créé une multitude de plugins et de loaders, il ne reste plus que de chercher et d'implémenter.

Avantages

- Permet d'utiliser Babel.js, outil permettant de compiler du code javascript ES5, ES6, ES7, ES8, ES9 ..., en polyfill afin que les vieux navigateurs n'ayant pas encore implémentés les nouvelles fonctionnalités soient compatibles avec les standards établis.
- Permet d'utiliser nativement le module d'import d'ES6 entre les fichiers javascript.
- Permet de minifier le html, Javascript et CSS.
- Permet de facilement intégrer un préprocesseur tel que Sass ou Less.
- Permet de manipuler les images.

- Permet de manipuler les fonts.
- Permet de rajouter automatiquement des préfixes CSS pour les différents navigateurs.
- Permet de créer des fichiers HTML automatiquement suivant certains templates que l'on a prédéfini au préalable.
- ...

Webpack est utilisé dans une multitude de librairies et de frameworks tel que

- React
- Angular
- Vue
- Meteor
- Ember
- Backbone
- Gatsby
- Nextjs
- Nuxtjs
- ...

Parmi les compagnies qui l'utilise, on retrouve

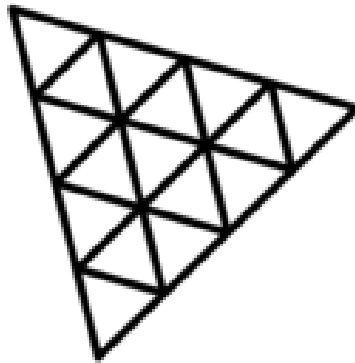
- Facebook
- Google
- Indeed
- Airbnb
- Shopify
- Discord
- Webflow
- Zeit
- Netlify
- CapitalOne
- Coinbase
- ...

Avec plus de 54'000 étoiles sur Github et des améliorations quotidiennes, c'est un outil qui est venu pour perdurer.

Uniquement les dépendances du globe virtuel seront compilées, les dépendances de développement dans le packages.json ne le seront pas.

Inconvénients

- Apprendre Webpack demande beaucoup d'investissement en recherches et temps. C'est un super outil, mais la réalité est que pas tout le monde n'est obligé de l'apprendre.
- Une seule personne implémentant webpack suffit pour un projet, en temps normal, il s'agit du chef de projet.
- Parfois, la configuration d'un projet en webpack est compliquée. Mais une fois configuré, on peut réutiliser notre configuration pour différents projets.



Threejs

Threejs est une librairie 3D permettant facilement d'interagir avec WEBGL, une api capable d'utiliser la carte graphique de l'utilisateur.

L'interaction entre threejs et WEBGL se fait via un langage nommé GLSL afin de calculer les millions de micros-interactions entre l'utilisateur et l'application.

Avantages

- Permet d'offrir à l'utilisateur une expérience 3D plaisante.

Inconvénients

- L'API de Threejs change très fréquemment, ainsi beaucoup d'applications ne fonctionnent plus lorsqu'elles mettent à jour la librairie.
- A l'heure actuel, soit le 01.04.2020, il y a environ plus de 456 bug reportés sur la page officielle de Threejs sur Github.
- Risque d'erreur accrue, pouvant provoquer une expérience utilisateur désagréable, puisque l'application tournerait au ralenti et n'utiliserait plus la carte graphique, dû aux erreurs lors de la mise à jour.

Communication

Architecture MVC(Model, View, Controller)

Dans une architecture de développement classique, le MVC est souvent utilisé pour mener à bien des projets.

Le modèle se traduit généralement par le modèle logique de données couplé à un ORM pour faciliter les interactions avec la base de données.

Le view est simplement le client, le frontend qui interagit avec le serveur.

Le controller est une action déclenchant une fonctionnalité précise côté serveur.

Exemple :

L'utilisateur navigue sur son site Internet préféré, il se connecte et peut commencer immédiatement à parler à ses amis.

Analyse :

L'utilisateur entre l'URL d'une page internet, le serveur répond à la requête de ce dernier, il lui renvoie une réponse sous forme de page HTML avec du CSS et Javascript.

Ensuite l'utilisateur se dirige sur la page se connecter, rentre ses informations. Une nouvelle fois le serveur répond à la requête, mais cette fois-ci, l'utilisateur désirant vouloir accéder une route précise, le serveur valide la requête à l'aide de controllers spéciaux, après vérification, il l'authentifie et lui donne des autorisations.

Le jeu continue pour le reste de la durée d'interaction avec le site web.

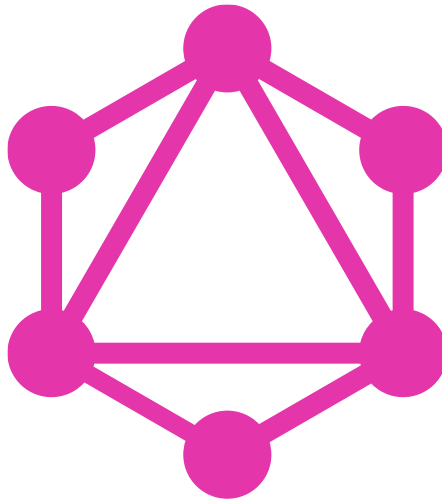
L'application du globe virtuel tournant sous Electron peut facilement utiliser l'architecture MVC, sous une autre forme, en utilisant une API Rest.

API Rest (Application Programming Interface)

Une API Rest une interface qui nous permet de communiquer avec des serveurs, objets connectés ou applications, via une interface de communication.

Une API Rest est exactement la même chose qu'un projet en architecture en MVC, à l'exception d'envoyer des pages web sous format HTML, CSS et Javascript, elle envoie des données, généralement, en format JSON ou XML.

C'est ensuite au tour du client de recevoir ces informations et de les utiliser pour afficher les informations et interagir avec.



GraphQL

GraphQL est une récente spécification voulant substituer REST.

Au lieu de définir des routes côté serveur pour accéder des informations précises à la base de données, c'est le client qui définit ce qu'il désire et le serveur lui envoie exactement ce que ce dernier a demandé.

Cela évite des larges payload pour une requête minime. Les soucis de over-fetching ou de under-fetching sont ainsi résolus.

- Dans le cas du globe virtuel, implémenter GraphQL permettrait d'effectuer qu'une seule requête pour obtenir toutes les images des enfants.
- Utiliser GraphQL serait fort intéressant, mais cela demande des connaissances techniques beaucoup plus poussées que ce qui est vu au CPNV.
- Pas nécessaire.



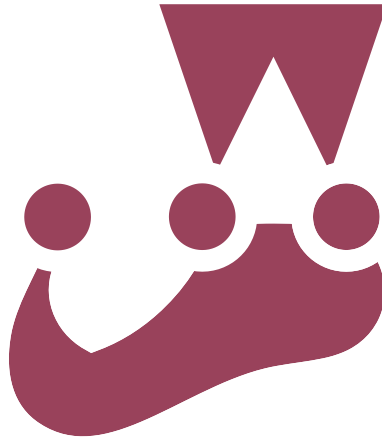
gRPC

gRPC est un framework rpc permettant d'effectuer des « remote procedure call » directement au serveur.

Utilisant fortement HTTP2, la connexion entre les deux parties est cryptée, permettant de garder de flux d'échange de données ouvert, à l'inverse d'une requête-reponse HTTP1.1, ou lorsque celle-ci se ferme, on est obligé d'effectuer une nouvelle requête pour obtenir une autre réponse désirée.

- Dans le cas du globe virtuel, implémenter HTTP2 côté serveur, permettrait de résoudre les milliers de requêtes effectuées pour obtenir toutes les images des enfants.
- Utiliser gRPC serait fort intéressant, mais cela demande des connaissances techniques beaucoup plus poussées que ce qui est vu au CPNV.
- Pas nécessaire.

Recommandations



Tests

Pour chaque projet, il est fortement conseillé d'effectuer des tests, notamment de tests unitaires et tests d'intégration au minimum afin de pouvoir assurer la fonctionnalité de ce dernier.

Pour le projet du globe virtuel, JEST fut ajouté à l'application ainsi qu'un script NPM a été défini pour lancer les tests.

Les fichiers tests se définissent par `nom.test.js`, ils sont automatiquement repérés par JEST.

JEST est une librairie de tests, d'assertion, de data mocking et de data coverage.

En effectuant souvent des tests, on évite des erreurs d'implémentations, pour la plupart, évitables.

- Je recommande fortement d'utiliser JEST afin d'assurer la qualité du projet.



TypeScript

TypeScript est un langage superset de Javascript, implémentant des types en développement, permettant une fois le projet fini, de compiler en javascript pour la production.

Pouvoir implémenter des types à des variables, ou même pouvoir créer des interfaces, permet de se protéger contre soi.

On évite ainsi la plupart des erreurs durant la compilation, et non durant l'exécution, sauf pour les erreurs logiques.

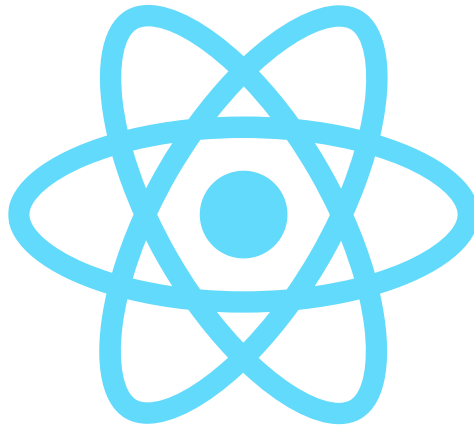
Utiliser TypeScript permet également de bénéficier de l'Intelligence mis en place par beaucoup d'IDE, soit des propositions en temps réel, afin d'améliorer la productivité des développeurs.

Grâce à TypeScript, on peut utiliser les dernières fonctionnalités de Javascript pas encore compatible dans tous les navigateurs, comme avec Webpack.

L'interopérabilité entre Javascript utilisant des fichiers TypeScript est maintenue.

Créé en 2012 par Microsoft, il est maintenu et amélioré tous les jours. Depuis, tous les grands codebases de libraires ou frameworks sont passés, pour la plupart, à du TypeScript. Y compris Three.js.

- Si le projet doit prendre de l'ampleur, utiliser TypeScript est fortement conseillé.
- Pour des petits projets, TypeScript n'est pas conseillé.



React

React est une librairie Javascript créé par Facebook en 2013, depuis avec la participation de la communauté, c'est devenu un écosystème.

C'est une librairie permettant de réutiliser des composants que l'on a préalablement créé, permettant de modifier ces derniers spécifiquement via un état mutable (state) ou via des propriétés (props).

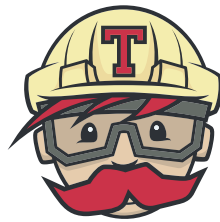
L'idée est de pouvoir réutiliser ce que l'on a créé, puisque dans une page il y a des éléments répétables tels que des cards, buttons, headers, footer, ..., pouvant être utilisé via différentes pages. Ainsi, copier et coller du HTML à travers différentes pages, n'est pas une solution.

Implémentant déjà webpack dans la librairie, il n'y a donc pas besoin de configurations pour le développeur, permettant une plus grande productivité.

Avec plus de 146'000 étoiles sur Github et des améliorations quotidiennes, c'est une librairie qui est venu pour perdurer.

Avec Angular et Vue, ce sont les meilleures librairies et framework frontend, actuellement, disponibles.

- Si le projet doit prendre de l'ampleur, utiliser React est fortement conseillé.
- Pour des petits projets, React n'est pas conseillé. Il y a une forte courbe d'apprentissage.



Travis CI

CI / CD

Continuous integration et continuous deployment, sont les standards du bon développement d'une application.

CI permet de récupérer le codebase, ensuite d'effectuer les tests s'il y en a, si tout se passe bien, le CD push l'application sur un serveur test ou sur un serveur de production que l'on aura au préalable défini.

Si le CI ne passe pas, l'utilisateur ayant commis le push reçoit une notification d'erreur.

A noter qu'aujourd'hui la relation entre CI / CD est devenue floue, la plupart des outils sur le marché permettent d'effectuer environ les mêmes fonctionnalités.

- Petit ou grand projet, je recommande fortement d'utiliser CI / CD, ça facilite énormément le développement.
- Outils gratuits recommandés : Travis CI, Circle CI



HEROKU

Heroku (Platform as a Service)

Heroku est un PaaS (Plateforme as a service) proposé par Salesforce, permettant d'héberger gratuitement un serveur avec base de données jusqu'à une certaine limite, après c'est payant.

Pour des projets test, nécessitant une base de données et un serveur, c'est un parfait sandbox.

Heroku permet d'utiliser Java, Golang, Nodejs, PHP, Python, Ruby, Scala et Closure comme langage côté serveur.

Il supporte comme base de données, MongoDB, Redis et Postgres.

L'intégration avec Github / GitLab est très simple.

- Très recommandé pour les petits projets. Eventuellement vérifier les prix d'Heroku afin de voir si ce n'est pas plus rentable, comparé à Infomaniak.
- Pour le plan gratuit, lorsque le serveur n'est pas utilisé, il se met en veille afin d'économiser du cpu, il faut 15-20s au serveur afin qu'il se réveille. A prendre en considération pour une API.



Netlify

Netlify permet d'héberger gratuitement un site web. Comparé à Heroku, il n'accepte que les fichiers HTML, CSS et Javascript.

Par défaut, il vient avec un CDN afin d'optimiser le téléchargement de fichiers.

-
- Très recommandé pour des petits projets.
-