

# Cours Web Mobile (CNAM, 2020)

Daniel VERMONDEN

6 avril 2020

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Comprendre les alternatives en matière de développement pour mobile . . . . .	2
1.2	Comprendre le concept des frameworks réactifs . . . . .	3
<b>2</b>	<b>Partie Web, avec Vue.js</b>	<b>4</b>
2.1	Installer un projet Vue.js simple . . . . .	4
2.2	Maîtriser les éléments de base du fonctionnement de Vue (directives et API) . . . . .	5
2.3	Elaborer un projet simple avec Vue . . . . .	6
2.4	Exploiter une API REST distante avec Vue . . . . .	7
2.5	Installer un projet Vue.js avec VueCLI et comprendre son architecture (npm, Webpack, etc.) . . . . .	9
2.6	Maîtriser le routage avec Vue.js . . . . .	10
2.7	Maîtriser les composants avec Vue . . . . .	11
2.8	Paramétrer et déployer sa première PWA avec Vue . . . . .	12
<b>3</b>	<b>Partie développement hybride avec Nativescript</b>	<b>13</b>
3.1	Expliquer ce qu'est Nativescript (et ses alternatives : Cordova, etc.)	13
3.2	Jouer avec NativeScript playground et découvrir les éléments d'interface . . . . .	14
3.3	Installer les outils pour développer avec Nativescript . . . . .	16
3.4	Gérer la navigation dans une application hybride . . . . .	16
3.5	Exploiter les API du téléphone avec Nativescript . . . . .	16

# **1 Introduction**

## **1.1 Comprendre les alternatives en matière de développement pour mobile**

### **Sujet projet**

A partir des ressources documentaires disponibles en ligne, rédiger un tableau de synthèse des alternatives possibles pour le développement d'applications sur mobile, en précisant pour chacune les technologies, le(s) langage(s), les avantages et les inconvénients.

Il s'agit bien de rédiger votre synthèse personnelle(par groupe) et non de reprendre intégralement un tableau tout fait sur Internet.

### **Pré-requis**

Aucun

### **Modalités (Travail en groupe ou travail individuel)**

Travail par groupe de 3.

### **Contraintes**

- Contrainte de temps : vous avez 2h00

### **Compétences visées par le projet**

- Comprendre les alternatives possibles pour le développement sur mobile, avec les avantages et les inconvénients de chacune.

### **Ressources**

- Internet (toutes les ressources documentaires en ligne)

### **Livrables**

- Un document de synthèse sous forme de tableau
- Une liste commentée des ressources documentaires utilisées

## 1.2 Comprendre le concept des frameworks réactifs

### Sujet projet

Pour le développement d'applications mobiles non natives (Web ou hybrides), on utilise le plus souvent aujourd'hui un "framework réactif". Les principaux frameworks réactifs sont Angular, React et Vue. Mais qu'est-ce que ces frameworks ont apporté de nouveau ?

Le fichier jquery-vue.html dans le dossier "comprendre-frameworks-reactifs" du dépôt Github du cours présente 2 alternatives de développement d'une mini-app : en jQuery d'une part et en Vue d'autre part.

Il s'agit de comprendre le code de ces mini-apps et d'identifier le changement majeur apporté par les frameworks réactifs dans le développement des applications.

### Pré-requis

- Maîtrise élémentaire de JavaScript

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

Aucune (mais il ne s'agit pas non plus d'y passer une journée complète;-) )

### Compétences visées par le projet

- Comprendre une mini-application en JavaScript
- Comprendre le changement majeur apporté par les frameworks réactifs

### Ressources

- 2 excellentes ressources sur JavaScript :
  - <https://javascript.info/>
  - <https://developer.mozilla.org/fr/docs/Web/JavaScript>

### Livrables

- Fichier de code source commenté, et avec en introduction une explication synthétique du changement majeur apporté par les frameworks réactifs.

## 2 Partie Web, avec Vue.js

### 2.1 Installer un projet Vue.js simple

#### Sujet projet

Un des atouts de Vue est sa documentation qui est, je trouve, très bien faite. Vérifions cela.

Pour commencer à prendre en main Vue, je vous propose de suivre la documentation pour installer Vue et créer vos premières mini-applications Vue.

Rendez-vous ici <https://fr.vuejs.org/v2/guide/#Pour-commencer> et créez dans un fichier html les 6 mini-apps proposées.

#### Pré-requis

Maîtrise de base de HTML, CSS et JS

#### Modalités (Travail en groupe ou travail individuel)

Travail individuel

#### Contraintes

Aucune

#### Compétences visées par le projet

— Etre capable d'installer une mini-app Vue

#### Ressources

La document en ligne de Vue, et particulièrement cette page : <https://fr.vuejs.org/v2/guide/#Pour-commence>

#### Livrables

Le code source des 6 mini-apps dans un fichier HTML, avec un commentaire sur les lignes qui incluent des directives Vue ou des propriétés de l'API de Vue.

## 2.2 Maîtriser les éléments de base du fonctionnement de Vue (directives et API)

### Sujet projet

Avant d'écrire vos propres applications avec Vue, il est utile de découvrir d'abord de petites applications ciblées. Les applications se trouvent dans le dépôt Github du cours (<https://github.com/dv-formateur/cnam-web-mobile-2020>), dans le dossier "exemples projets Vue simples".

L'objectif ici est d'utiliser chacune des applications suivantes, puis de lire leur code et de le comprendre. Vous commenterez les lignes qui concernent des éléments de Vue.

### Pré-requis

- Maîtriser l'installation de base d'une application Vue.
- Maîtrise de base de JavaScript

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

Aucune

### Compétences visées par le projet

Maîtriser les principales directives de Vue, ainsi que les principales propriétés de son API.

### Ressources

- Document de Vue : <https://fr.vuejs.org/v2/guide/>
- Les ressources JavaScript déjà mentionnées plus haut.

### Livrables

Le code source des applications agrémenté de vos commentaires, sur votre compte Github, dans un dépôt dédié nommé "vue-apps-commented", avec votre pseudo dans le Readme.

## 2.3 Elaborer un projet simple avec Vue

### Sujet projet

Il s'agit maintenant de créer votre première application Web mobile "from scratch". A ce stade, on ne va pas recourir à une API REST distante (ce sera la prochaine étape).

Le thème de cette application sera discuté ensemble. Voici quelques idées de fonctionnalités possibles :

- formulaire avec questions spécifiques en fonction de réponses préalables,
- affichages alternatifs,
- filtre de recherche.

### Pré-requis

- Maîtrise du HTML et du CSS
- Maîtrise de base de JavaScript
- Maîtrise de base des directives de Vue et des propriétés de son API

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

- Le code doit être le vôtre, pas une copie de code trouvée sur le Web - ceci n'empêche pas évidemment de reprendre des bouts courts de code proposés sur les forums d'entraide pour telle ou telle fonctionnalité.
- L'interface doit être développée selon l'approche responsive "Mobile First".
- Votre code doit être commenté.

### Compétences visées par le projet

Etre capable de développer une application simple avec Vue, en autonomie

### Ressources

- Documentation JavaScript
- Documentation de Vue
- Document MDN

### Livrables

- Le code source de l'application sur Github, dans un dépôt "vue-app-no-api"
- Le code source doit comporter des commentaires explicatifs
- L'application doit être déployée en ligne. Le lien vers l'application est disponible dans le Readme du dépôt Github "vue-app-no-api", avec votre pseudo dans le Readme.

## 2.4 Exploiter une API REST distante avec Vue

### Sujet projet

On avance dans la maîtrise des applications Web mobiles avec Vue. On va maintenant s'intéresser à une fonctionnalité indispensable des applications de manière générale et des applications Web en particulier : la récupération de données depuis une API REST distante.

Il s'agit pour ce mini-projet de mettre au point une nouvelle application 'from scratch' qui exploite une API distante.

Vous avez la possibilité de me proposer une API REST que vous voudriez exploiter. A défaut, je vous invite à exploiter les données libres publiques disponibles via les portails opendata, par exemple :

- <https://www.data.gouv.fr/fr/>
- <https://data.nantesmetropole.fr/pages/home/>

Github offre également une API REST intéressante.

Le choix de l'API REST retenue devra être préalablement validé avant de vous lancer dans le développement de l'application.

### Pré-requis

- Maîtrise du HTML et du CSS
- Maîtrise de base de JavaScript
- Maîtrise de base des directives de Vue et des propriétés de son API

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

- Le code doit être le vôtre, pas une copie de code trouvée sur le Web - ceci n'empêche pas évidemment de reprendre des bouts courts de code proposés sur les forums d'entraide pour telle ou telle fonctionnalité.
- L'interface doit être développée selon l'approche responsive "Mobile First".
- Votre code doit être commenté.

### Compétences visées par le projet

Etre capable de récupérer des données distantes d'une API REST et de les exploiter dans une application Vue.

### Ressources

- Documentation JavaScript
- Documentation de Vue
- Document MDN

Une requête API en Javascript peut être réalisée avec `xmlhttprequest` (mais l'API n'est pas très intuitive), avec `fetch` (l'API est plus intuitive), ou bien avec la librairie externe `Axios` (plutôt pratique à utiliser).

- API `xmlhttprequest` : <https://developer.mozilla.org/fr/docs/Web/API/XMLHttpRequest>

- API fetch : [https://developer.mozilla.org/fr/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/fr/docs/Web/API/Fetch_API)
- Axios : <https://github.com/axios/axios>
- Un article : <https://medium.com/javascript-in-plain-english/comparing-different-ways-to>

### **Livrables**

- Le code source de l'application sur Github, dans un repository "vue-app-with-rest-api", avec votre pseudo Github dans le Readme



## 2.5 Installer un projet Vue.js avec VueCLI et comprendre son architecture (npm, Webpack, etc.)

### Sujet projet

Quand on veut développer une application un peu significative avec Vue, on n'utilise généralement pas le simple script de la Vue dans une page Web. On utilise plutôt Vue en tant que package importé depuis NPM, et on combine cela avec un outil de build comme Webpack.

Cette architecture complique sensiblement l'installation mais elle offre aussi une série d'avantages (heureusement !). Comme on le verra dans un autre mini-projet, cela simplifie notamment le travail à plusieurs, en élaborant des "composants à fichier unique" ("single file components"). Webpack permet aussi de réaliser une série d'opérations pour "construire" l'application (phase de build). En outre, on s'épargne aussi le besoin de recharger la page à chaque modification.

Vue offre un merveilleux petit outil pour mettre en place des applications avec cette architecture : VueCli.

Votre mission est d'installer VueCli, puis de mettre en place un projet avec cet outil, en vous interrogeant sur les différentes options proposées et sur les différents outils mis en place.

### Pré-requis

- Maîtrise du HTML et du CSS
- Maîtrise de base de JavaScript
- Maîtrise de base des directives de Vue et des propriétés de son API

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

Aucune

### Compétences visées par le projet

- Installer un projet avec VueCli
- Maîtriser les différentes options proposées
- Comprendre les différents outils associés : Webpack, package.json, etc.

### Ressources

- Documentation de VueCli : <https://cli.vuejs.org/>

### Livrables

Une courte vidéo de votre installation en local, présentant le processus de création d'un nouveau projet, avec le commentaire des options choisies.

## 2.6 Maîtriser le routage avec Vue.js

### Sujet projet

Avec notre installation "complète" de Vue, nous avons pu constater l'existence d'un routage, avec notamment l'exemple de la route `"/about"`. Dans ce mini-projet, nous allons découvrir de manière plus approfondie ce routeur, son fonctionnement et ses options.

Cette mission consiste à créer un espace protégé sur notre application accessible avec un login et un mot de passe. Il s'agit aussi de protéger les routes associées si on essaie d'y accéder directement depuis la barre d'adresse du navigateur.

Avancé : créer une route dynamique

TODO Par exemple, avec appli précédente, ajouter `/documentation`, `/cgv`, etc.

### Pré-requis

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

- Le code doit être le vôtre, pas une copie de code trouvée sur le Web - ceci n'empêche pas évidemment de reprendre des bouts courts de code proposés sur les forums d'entraide pour telle ou telle fonctionnalité.
- Votre code doit être commenté.

### Compétences visées par le projet

Maîtriser le routage avec Vue.

### Ressources

- La document de référence : <https://router.vuejs.org/fr/>

### Livrables

- Le code du projet sur le dépôt TODO LEQUEL
- TODO AUTRES ?

## **2.7 Maîtriser les composants avec Vue**

### **Sujet projet**

Il s'agit dans cette mission de développer un composant.  
A PRECISER

### **Pré-requis**

- Maîtrise de base de l'organisation des fichiers dans un projet installé avec VueCli
- Maîtrise de base de l'API et des directives de Vue
- Maîtrise de JavaScript

### **Modalités (Travail en groupe ou travail individuel)**

Travail individuel

### **Contraintes**

- Ecrire son propre code !

### **Compétences visées par le projet**

- Maîtriser les "single file components"

### **Ressources**

A COMPLETER

### **Livrables**

- Le code du projet sur le dépôt sur le dépôt Github nommé "vue-app-sfc", avec votre pseudo dans le Readme, ainsi qu'une adresse de déploiement du projet.

## **2.8 Paramétrer et déployer sa première PWA avec Vue**

A PRECISER

## 3 Partie développement hybride avec Nativescript

### 3.1 Expliquer ce qu'est Nativescript (et ses alternatives : Cordova, etc.)

#### Sujet projet

L'objectif ici est de se documenter sur NativeScript, de comprendre son objectif et de situer cette solution dans son paysage technique (quelles sont les alternatives pour mettre au point des applications hybrides, avec Vue.js, mais aussi sans Vue.js).

#### Pré-requis

Aucun

#### Modalités (Travail en groupe ou travail individuel)

Par groupe de 3

#### Contraintes

Ne pas y passer une journée complète ;-)

#### Compétences visées par le projet

- Comprendre l'objectif de NativeScript
- Connaître les alternatives à NativeScript

#### Ressources

Notamment,

- <https://stackoverflow.com/questions/53867122/which-is-better-vue-native-or-nativescript>
  - <https://medium.com/learning-lab/lessons-learned-on-writing-apps-with-nativescript-vu>
- mais ne vous limitez pas à ces ressources !

#### Livrables

- Un document de synthèse
- Une liste commentée des ressources documentaires utilisées

## 3.2 Jouer avec NativeScript playground et découvrir les éléments d'interface

### Sujet projet

NativeScript a l'avantage de proposer une interface en ligne avec quelques applications fonctionnelles (playground). Cela permet une première exploration de ce framework, sans devoir au préalable devoir installer les outils associés, à l'exception de l'outil de preview.

Votre mission consiste à choisir une des applications exemple (utilisant Vue) et à en explorer et comprendre le fonctionnement, notamment l'organisation des fichiers et l'élaboration des interfaces.

### Pré-requis

- Maîtrise de JavaScript
- Maîtrise de Vue.js

### Modalités (Travail en groupe ou travail individuel)

Travail individuel

### Contraintes

Aucune

### Compétences visées par le projet

- Comprendre les spécificités de l'élaboration d'applications hybrides avec NativeScript, particulièrement la spécificité de la mise au point des interfaces par rapport au Web.

### Ressources

Une présentation du Playground : <https://nativescript-vue.org/en/docs/getting-started/playground-tutorial/>

Documentation NativeScript <https://docs.nativescript.org/start/introduction>

Documentation NativeScript Vue <https://nativescript-vue.org/en/docs/getting-started/installation/>

Liste des applications de démo : [https://market.nativescript.org/?tab=samples&framework=vue&category=all\\_samples](https://market.nativescript.org/?tab=samples&framework=vue&category=all_samples)

Sur les éléments de layouts : <https://docs.nativescript.org/ui/layouts/layout-containers>

Sur les composants : <https://docs.nativescript.org/ui/components/action-bar>

Sur la gestion d'éléments spécifiques par plateforme : <https://dev.to/progress/platform-specific-development-with-nativescript-dd1>

Vidéo : <https://www.youtube.com/watch?v=iGa1HQbtDmU&list=PL3TJnCmVR-22RKxnis9cDAJSciJcOV3>

Présentation de NativeScript et NativeScript Vue [https://www.youtube.com/watch?v=claDp19\\_aqA&list=PL3TJnCmVR-22RKxnis9cDAJSciJcOV3wf&index=14](https://www.youtube.com/watch?v=claDp19_aqA&list=PL3TJnCmVR-22RKxnis9cDAJSciJcOV3wf&index=14)

14

## **Livrables**

Fournir une note de synthèse expliquant

- la structure des fichiers, et
- le code pour la mise au point des interfaces

### **3.3 Installer les outils pour développer avec Nativescript**

A PRECISER

### **3.4 Gérer la navigation dans une application hybride**

A PRECISER

### **3.5 Exploiter les API du téléphone avec Nativescript**

A PRECISER