

React Workshop

Ce workshop permet de découvrir [react](#) et son écosystème par la pratique, étape par étape !



react-workshop de [Mathieu ANCELIN et Sébastien PRUNIER](#) est mis à disposition selon les termes de la [licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#).

Le sujet

Lors de ce workshop, nous allons développer une application web permettant de gérer ses vins préférés* !

Les principales fonctionnalités de l'application sont :

- Lister les vins par région viticole,
- Afficher la fiche détaillée d'un vin,
- Aimer un vin,
- Ajouter un commentaire sur un vin.

** L'abus d'alcool est dangereux pour la santé, à consommer avec modération ;-)*

Attention

Durant votre voyage initiatique au coeur de `react` vous allez vous rendre compte, avec épouvante, que malgré de nombreuses descriptions des choses à faire dans les étapes, certains morceaux de code sont déjà écrit !!! Ceci est bien évidemment volontaire de notre part pour afin de vous laisser un maximum de temps pour vous consacrer aux choses importantes et de ne pas vous ennuyer outre mesure avec des tâches fastidieuses.

Si cette philosophie ne vous convient pas, vous pouvez également démarrer votre application de gestion des vins de zéro dans un dossier à part et suivre les indications dans les diverses étapes.

Les pré-requis techniques

Les pré-requis techniques sont les suivants :

- Node (version 4.x ou 5.x) et NPM (version 3.x)
- Git
- Atom
- React Developer Tools

La dernière étape de ce workshop implique l'utilisation de `react-native` . En plus des outils précédant vous aurez besoin de :

- Toutes plateformes
 - `watchman`
 - `react-native-cli`

- iOS
 - Xcode
- Android
 - Java (JDK8)
 - Gradle
 - Android Studio

Node.js

Téléchargez et installez la version de Node.js correspondant à votre système d'exploitation, en suivant les indications disponibles sur le site officiel :

<https://nodejs.org/en/download/>

Vérifiez l'installation en lançant les commandes suivantes dans un terminal :

```
$ node -v
v4.4.0

$ npm -v
3.8.1
```

Si npm n'est pas en version 3.x, vous pouvez effectuer la mise à jour grâce à la commande suivante : `npm upgrade -g npm`

Attention: il semblerait que le workshop fonctionne bien avec Node 0.12.x et NPM 2.x, vous pouvez donc conserver votre setup si besoin. Mais sérieusement, Node 0.12 ???

Git

Téléchargez et installez la version de Git correspondante à votre système d'exploitation, en suivant les indications disponibles sur le site officiel : <https://git-scm.com/downloads>

Vérifiez l'installation en lançant la commande suivante dans un terminal :

```
$ git --version
git version 2.7.3
```

Atom

L'éditeur préconisé pour le workshop est [Atom](#).

Téléchargez et installez Atom, puis installez les packages suivants :

- language-javascript-jsx
- linter-eslint

Pour savoir comment gérer les packages d'Atom : <https://atom.io/docs/latest/using-atom-atom-packages>

React Developer Tools

Afin de disposer d'outils spécifiques à `react` dans votre navigateur web, installez **React Developer Tools** :

- [React Developer Tools pour Google Chrome](#)
- [React Developer Tools pour Mozilla Firefox](#)

Xcode

Xcode est installable depuis le Mac App Store présent sur votre Mac

Vous pouvez l'installer directement depuis [ici](#)

Java Development Kit

Si vous n'avez pas de JDK 8 installé, rendez-vous sur le [site d'Oracle](#) pour télécharger et installer la dernière version du JDK.

Attention, ce n'est pas parce que vous avez Java sur votre machine que vous avez le JDK. Testez avec la commande `javac`

Gradle

Gradle est un outil de build utilisé par les projets Android. Vous pouvez procurer la dernière version de Gradle [ici](#)

Android Studio

Android Studio est l'IDE officiel pour développer des applications Android. Vous pouvez télécharger une version du studio [ici](#).

Il vous faudra ensuite configurer un émulateur Android.

Pour cela, vous pouvez suivre [cette documentation](#) sur le site de `react-native`.

En ce qui concerne l'émulateur, si vous n'avez rien d'existant pour le moment, vous devez suivre [cette partie de la documentation](#) en utilisant le nom d'émulateur `reactnative` et en utilisant `1Go` de mémoire interne et `1Go` de mémoire externe et `2Go` de RAM, cf. [cette capture d'écran](#)

watchman

Sous Linux et Mac OS, vous aurez sûrement besoin d'installer `watchman`, suivez les [instructions d'installation](#).

react-native-cli

```
npm install -g react-native-cli@0.2.0
```

Pré-installer les dépendances

Pour pré-installer les dépendances NPM à l'avance, vous pouvez lancer le script `install.sh` qui lancera les commandes `npm install` dans les différentes étapes du workshop.

Nous vous recommandons également de télécharger un exemplaire des dépendances pour éviter les problèmes de réseau potentiels

- [node modules pour Mac OS](#)
- [node modules pour Linux](#)
- [node modules pour Windows](#)

Pour la partie `react-native`

- [node modules pour Mac OS](#)
- [node modules pour Linux](#)
- [node modules pour Windows](#)

Si les dépendances posent des problèmes, notamment au niveau des paquets natifs, n'hésitez pas à lancer un `npm rebuild` pour recompiler ces paquets sur votre machine.

API

L'application de gestion des vins utilise une API REST comme source de données lui permettant d'afficher des vins par régions, avec leur détail, une photo de la bouteille, etc ... Cette api est en fait un petit serveur NodeJS/Express avec des données en mémoire disponible dans le dossier [api](#). Pour toutes les étapes du workshop vous allez avoir besoin de cette API pour alimenter votre application. Vous pouvez donc déjà démarrer le serveur d'API dans un onglet à part de votre terminal afin qu'il soit toujours disponible.

Pour démarrer le serveur exposant l'API, lancez les commandes suivantes :

```
$ cd api
$ npm install
$ npm start
```

Rendez-vous ensuite sur <http://localhost:3000> pour parcourir la documentation des différentes routes disponibles. Nous vous recommandons de passer au moins une fois sur cette documentation afin d'avoir une idée globale des données fournies par l'API pour alimenter votre application.

Tests

De manière générale, chaque étape du workshop (hormis les premières) possède des tests pour valider le fonctionnement de l'application. Ces tests vont vous permettre de savoir ou vous en êtes dans l'étape courante. N'hésitez pas à lire les tests pour voir de quelle façon l'application doit se comporter, ainsi qu'à lancer la commande magique (`npm test`) à tout moment.

Les étapes du workshop

- [Etape 0](#)
 - Mise en place des outils de build
 - webpack
 - webpack-dev-server
 - babel
 - eslint
 - Création du premier composant `react`
- [Etape 1](#)
 - Ajout de tests unitaires sur le composant créé à l'étape 0
 - react-test-utils
 - mocha
 - chai
 - jsdom
- [Etape 2](#)
 - Création d'une application `react` basique

- Découpage en composants
 - Interactions entre composants
 - Utilisation du `state`
- Utilisation de l'API (Appels AJAX)
 - Liste des régions viticoles,
 - Liste des vins d'une région
 - Détail d'un vin
- Bonnes pratiques `react`
 - PropTypes
 - "Dumb components" vs "Smart components"
- [Etape 3](#)
 - Single Page Applications
 - Introduction à `react-router`
 - Refactoring de l'étape 2 pour obtenir une SPA
- [Etape 4](#)
 - Ajout de fonctionnalités
 - Aimer un vin
 - Ajouter un commentaire
- [Etape 5](#)
 - Introduction à redux
 - Ajout d'une fonctionnalité avec redux
- [Etape 6](#)
 - Refactoring complet de l'application avec redux
- [Etape 7](#)
 - Nouveaux patterns `react`
 - Classes
 - HOC
 - Stateless components
 - Contexts
- [Etape 8](#)
 - `react-native`
 - Implémentation de l'application en `react-native`

A vous de jouer !

C'est le moment de commencer, si vous souhaitez en savoir plus sur la façon de monter un projet `react`, installer le tooling, etc ... vous pouvez partir de [l'étape 0](#) sinon vous pouvez directement vous rendre à [l'étape 1](#). Bon courage !!!