

[← My path](#) PROJECT IN PROGRESS

## Faites passer une librairie jQuery vers React

[PROJECT BRIEF](#)[RESOURCES](#)[ASSESSMENT](#) 110 hours

Last updated on Wednesday, September 2, 2020

Vous travaillez pour le département technologique d'une grande société financière, WealthHealth.

Cette société utilise une application web interne, appelée HRnet, qui gère les dossiers des employés.

L'application est ancienne et utilise jQuery côté front end, ce qui entraîne des bugs considérables et une augmentation des plaintes en interne. Votre équipe de développement s'efforce depuis un certain temps déjà de mettre à niveau l'application.



WEALTH HEALTH

Logo de Wealth Health

Un matin, vous vous rendez au travail et remarquez les messages suivants de Jade, une collègue développeuse de l'équipe :

**Jade** : Salut, toi ! J'espère que tu as passé un bon week-end ! Tu as entendu la nouvelle ? La direction va ENFIN nous laisser mettre à jour HRnet et le convertir en React ! Plus de maintenance du vieux code jQuery !! Youhouuu !! 🎉🌐

**Vous** : Oh, vraiment ? Trop bien ! Le jour est enfin arrivé ! 😱🎉

**Jade** : Oui, c'est tellement excitant ! Et puisque tu t'es donné à fond et que tu as fait un travail incroyable ces derniers temps, tu seras l'un des premiers développeurs à commencer à convertir le code ! Félicitations !

**Vous** : C'est génial ! J'ai hâte de me mettre dedans pour résoudre (enfin) ce problème ! HRnet à lui seul doit représenter environ 90% de notre dette technologique totale, lol 😂

**Jade** : M'en parle pas ! Je me suis dit que je t'enverrais des informations pour t'aider à démarrer. Comme tu le sais sûrement, les plus gros problèmes pour les utilisateurs de HRnet en ce moment sont les sélecteurs de date, les fenêtres modales, les menus déroulants et les tableaux. Nous avons reçu plusieurs plaintes disant que ces plugins jQuery sont très lents.

Nous voulons donc créer nos propres composants React à la place de ces plugins jQuery tiers qui sont utilisés dans l'interface utilisateur. Et on espère que la conversion de ces plugins jQuery en composants React améliorera les performances et la stabilité.

**Vous** : Ok, compris. Tous ces appels réseau et toute cette manipulation DOM nécessaire pour les plugins jQuery s'additionnent et ralentissent les choses. Et ce sera bien de commencer à avoir nos propres composants React pour avoir plus de contrôle sur l'application.

**Jade** : Exactement ! La direction ne veut pas que cette première partie du processus de conversion prenne trop de temps, alors s'il te plaît, choisis UN SEUL des quatre plugins jQuery et convertis-le en composant React. (Je te les envoie plus tard dans la journée). Tu devras créer un repo GitHub séparé pour le code converti.

**Vous** : Bien sûr ! C'est une bonne manière d'avoir une version initiale de la nouvelle application plus rapidement, et nous pourrons toujours itérer dessus plus tard.

**Jade** : Exactement ! Pour tester manuellement tes composants React convertis, tu devras aussi convertir l'application HRnet principale en une application React. Comme tu sais que jQuery et React ne vont pas bien ensemble, nous voulons donc que la nouvelle version de l'application soit 100% React et 0% jQuery. Nous ne voulons pas créer un genre de monstre jQuery/React !

**Vous** : C'est compris. Merci de le préciser, je commençais déjà à trembler à l'idée de créer un hybride React/jQuery 😱

**Jade** : Non 🤪 On ne veut vraiment pas ça ahah

Je t'enverrai le repo actuel de HRnet pour que tu puisses commencer. Nous aurons aussi besoin que tu fasses des tests de performance et que tu nous fournisses des rapports.

**Vous** : Bien vu ! Je veillerai à inclure ces rapports de performance dans mes livrables.

**Jade** : Aussi, garde à l'esprit que nous voulons suivre un paradigme de programmation fonctionnel en écrivant ces bibliothèques dans React, donc évite d'utiliser des classes quand tu le convertis. Et essaie d'écrire des morceaux de code plus petits et modulaires et des fonctions autonomes pour une modularité et une maintenabilité optimales.

Aussi, assure-toi de documenter ton composant React converti avec une description générale de ce que fait le composant et des commentaires expliquant à quoi chaque accessoire est destiné et comment il est utilisé. Le reste de l'équipe te remerciera ! 😊

**Vous** : D'accord, je m'occupe de ça !

**Jade** : Perfecto ! Cela devrait être tout ce dont tu as besoin pour commencer - je t'envoie aussi un e-mail récapitulatif tout à l'heure. Si tu as des questions ou si tu as besoin d'autres détails ou ressources, n'hésite pas à me le faire savoir. Allons nous débarrasser de ce code jQuery ! 🎉

**Vous** : Je suis sur le coup ! Tu peux compter sur moi, Jade ! 😁

Comme promis, un peu plus tard dans la journée, Jade vous envoie le mail suivant :

**Objet:** Détails sur le projet de refonte

**De:** Jade

**À:** Moi

Salut à toi !

J'ai hâte que tu puisses démarrer ce projet ! Voici un récapitulatif des principales tâches :

- Convertir l'ensemble du projet HRNet en React.
- Convertir l'un des quatre plugins jQuery actuels en React. Remplacer les 3 plugins jQuery restants par des composants React.
- Effectuer des tests de performance Lighthouse en comparant l'ancienne et la nouvelle application.

Tu trouveras toutes les informations clés pour chaque tâche ci-dessous :

### Conversion du projet HRNet

Voici [le repo HRnet](#) actuel. N'oublie pas que toute l'application HRNet doit être convertie en React :

- Tu devras faire une nouvelle version des pages "Create Employee" et "Employee List" avec React.
- Tu devras ajouter un système de gestion d'état (la version actuelle utilise un stockage local).
- Tu dois aussi t'assurer que tout est cohérent au niveau du style. Pour cela, tu n'es pas obligé de refaire le design de l'application, mais si tu veux changer le style pour quelque chose de plus moderne, tu es le bienvenu.
- Si tu as le temps, tu peux tester le code React avec une suite de tests unitaires. Sinon, seuls des tests manuels sont nécessaires.

### Conversion des plugins

Pour les plugins, n'hésite pas à consulter l'onglet Issues pour avoir plus de contexte sur les problèmes que les utilisateurs rencontrent avec les plugins jQuery existants : [issue sur les sélecteurs de date](#), [issues de fenêtres modales](#), [issue sur menus déroulants](#), et [issue sur les tableaux](#).

Voici la liste des plugins jQuery actuellement utilisés qui doivent être convertis :

- [Plugin de sélection de date](#)
- [Plugin de fenêtre modale - jQuery.modal.js](#)
- [Menus déroulants](#)
- [Plugin pour les tables de données](#)

De plus, lors de la conversion d'un plugin jQuery en un composant React, il faut garder à l'esprit de ne convertir que le code qui traite de la fonctionnalité réelle de l'interface utilisateur du plugin. Par exemple, si un plugin jQuery inclut du code AJAX, tu n'as pas besoin de le convertir. Si tu convertis un plugin jQuery pour une fenêtre modale, concentre-toi sur la création d'un composant React qui fonctionne comme une fenêtre modale, et rien d'autre.

### Tests de performance

Nous voulons également mesurer des données quantifiables (ex. : temps de chargement des pages, appels réseau) pour nous assurer que la conversion de l'application à React améliore effectivement les performances. Pour cela, fais bien des audits de performance Lighthouse. Pour comparer, tu devras en faire un pour l'application jQuery HRnet actuelle, puis un autre une fois que l'application et le plugin jQuery choisi seront convertis en React.

C'est tout ! Une fois que l'application HRnet en React fonctionne, tu peux publier le composant React sur npm sous forme de package et partager le lien pour que nous puissions l'utiliser si nécessaire. Si tu as du mal avec npm, tu peux utiliser les paquets GitHub comme alternative.

Aussi, fais-moi savoir quand tout est fait et nous ferons une révision du code pour répondre aux commentaires restants.

J'ai hâte de voir ce que tu vas faire !

**Jade**

Ça y est, vous avez toutes les informations dont vous avez besoin. Maintenant, le plus dur sera de choisir quel plugin jQuery convertir !

## Livrables

- Un **repo GitHub** contenant :
  - le code source du plugin jQuery sélectionné converti en ReactJS (un seul plugin doit être converti).
  - Un lien vers la bibliothèque React convertie publiée sur npm (ou GitHub Packages) dans le fichier README.
- Un **second repo GitHub** contenant le code source de l'application HRnet converti en React.
- Un **rapport de performance LightHouse** de HRnet fonctionnant avant et après la conversion de la bibliothèque, au format JSON.

Le repo GitHub doit être nommé avec la convention suivante : Nomcomplet\_#\_Datedémarrage. Le # correspond au numéro du projet sur le parcours et la date doit être au format jjmmaaaa. Par exemple, FrancoisLenotre\_12\_05032020.

## Soutenance

La présentation sera structurée comme suit, l'évaluateur jouant le rôle de Jade :

- Présentation et explication des livrables (10 minutes)
- Discussion (15 min)
  - L'évaluateur vous posera des questions sur votre méthodologie et vos résultats.
  - Soyez prêt à défendre votre travail, car l'évaluateur mettra en question vos décisions.

L'évaluateur cessera de jouer le rôle de Jade cinq minutes avant la fin de la session afin que vous puissiez faire un compte rendu ensemble.

## Skills



Refondre une application pour réduire la dette technique



Analyser la performance d'une application web



Déployer une application front-end



---

OPENCLASSROOMS



---

OPPORTUNITIES



---

SUPPORT



---

FOR BUSINESS



---

MORE



English ▾



