**exo-react-2023**

Vous commencez le point de contrôle à partir de zéro ! Il faut donc tout installer. Vous pouvez soit utiliser : [create-react-app](https://create-react-app.dev/)

**1️⃣ React**

**Dans cette étape, vous allez créer une web-app en utilisant React, qui affiche les jeux de l'une de nos API !**

S'il vous plait, n'oubliez pas de faire des commits atomiques avec des messages explicites 🙏

Vous pouvez y trouver la documentation de l'API :

<https://apis.wilders.dev/wild-games>

**Point de départ**

* Vous pouvez effacer tout le code passe-partout jsx dans App.jsx
* Ensuite, créez un composant <Header /> qui affiche un message de bienvenue sympa.
* Passez un nom de props à ce composant, pour afficher le nom de votre site Web (réagissez au jeu si vous n'avez aucune idée)

*commit*

**Une liste ?**

* Votre prochaine mission est d'afficher la liste complète des jeux.
* Commencez par créer deux nouveaux composants <GameList /> et <Game />.
* Dans le composant <GameList />, vous devrez obtenir la liste de l'API et la mapper dans votre fichier jsx. Pour chaque jeu obtenu à partir de l'API, affichez un composant <Game />.
* Votre composant <Game /> aura besoin de quelques props pour afficher les informations du jeu. Il y a beaucoup d'informations que vous pouvez utiliser (nom, note, images, ...). Affichez-les au format de votre choix !

*commit*

**Filtration**

* Maintenant que vous affichez tous les jeux, votre prochaine mission est d'ajouter un filtre sur cette liste.
* Ajoutez un button au composant <GameList />, lorsque vous cliquerez dessus, il n'affichera que les jeux dont la note est supérieure à 4,5. En cliquant à nouveau sur ce bouton, la liste devrait afficher à nouveau tous les jeux.

*encore une fois, n'oubliez pas de commit* 😉

**Page de détails**

Vous allez utiliser react-router et react-router-dom

* Refactorisez votre code pour implémenter un routeur, avec deux routes :
* "/" affichera votre <GameList />
* "/games/:id" affichera le prochain composant que vous allez créer
* Créez un composant <GameDetails /> et utilisez-le dans votre route /games/:id.
* Ensuite, vous pouvez ajouter un lien (ou bouton) de navigation à votre composant <Game /> qui redirigera vers la page de détails de ce jeu.
* Le composant <GameDetails /> devra récupérer les données de l'API et afficher des informations détaillées sur le jeu en cours (ce que vous choisissez d'afficher dépend de vous !)

*tu sais quoi faire, non ? commiy 😉*

**Bravo ! Tu l'as fait ! 🎊**

**Bonus**

S'il vous reste du temps, voici quelques bonus. Comme ce sont des bonus, ils sont totalement facultatifs et vous ne devriez pas les faire si vous n'avez pas terminé les étapes précédentes.

**Suppression d'un jeu**

<Game /> contient un <button> qui permet de supprimer un jeu de la liste (il n'est pas nécessaire de le supprimer dans l'API).

**Utiliser un framework CSS**

Intégrez le framework CSS de votre choix dans votre application React. Si vous ne savez pas lequel choisir, nous pouvons vous recommander Tailwind.

Vous pouvez commencer par regarder la documentation sur la façon de l'installer.

*Faire attention ! ⚠️ Si vous choisissez de commencer à utiliser create-react-app, vous devrez installer CRACO, mais toutes les étapes sont expliquées ici.*