

ATELIER PRATIQUE

Initiation react 16.9 Hook

Mini projet avec l'API Swappi

Mini projet mon portfolio/book

Tests avec Jest



Compétences :

Activité type 1 « Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité » :

C4 : Développer la partie front-end d'une interface utilisateur web

Critères de performance

- L'interface est conforme à la maquette de l'application et au dossier de conception technique
- Les bonnes pratiques de développement web sont respectées
- Les règles d'accessibilité sont respectées
- Les pages web sont accessibles depuis un navigateur
- Les pages web s'adaptent à la taille de l'écran et sont fluides
- Le code source est documenté ou auto-documenté
- Les tests garantissent que les pages web répondent aux fonctionnalités décrites dans le dossier de conception technique
- Les tests de sécurité suivent un plan reconnu par la profession
- L'objet de la recherche est exprimé de manière précise en langue française ou anglaise
- La documentation technique liée aux technologies associées, en français ou en anglais, est comprise (sans contre-sens, ...)
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité
- La veille sur les vulnérabilités connues permet d'identifier et corriger des failles potentielles
- Le partage du résultat de veille est effectué oralement ou par écrit avec ses pairs

Savoir-faire techniques, savoir-faire organisationnels, savoir-faire relationnels, savoirs

Coder les pages web statiques et adaptables avec un langage de balise et feuilles de style, en suivant les principes de qualité et dans un style défensif en veillant à rendre l'application la plus résiliente possible

Utiliser un environnement de développement y compris en anglais

Publier l'interface web sur un serveur

Écrire un algorithme

Intégrer dans une page web des scripts événementiels avec un langage de script client, en suivant les principes de sécurisation des clients web

Utiliser un framework adaptatif

Programmer avec un framework de script client

Fluidifier le site en utilisant des mécanismes asynchrones (de type AJAX, ...)

Documenter les scripts client

Réaliser un jeu de tests des scripts client

Tester la sécurité du site avec un guide de test reconnu

Planifier et suivre les tâches de développement

Rechercher une réponse pertinente à une difficulté technique

Rechercher des informations sur des sites Internet, des forums et des FAQ francophones ou anglophones

Contribuer à la mise à jour des bases de connaissances francophones ou anglophones accessibles par Internet

Connaissance d'un environnement de développement

Connaissance du vocabulaire anglais professionnel associé au web

Connaissance de l'architecture du web et des standards de l'organisme de normalisation W3C

Connaissance des bonnes pratiques de qualité et d'éco conception web

Connaissance des langages du développement web, tels que langage de balise et feuilles de style

Connaissance d'un framework de présentation adaptatif

Connaissance des règles d'accessibilité des contenus web du type WCAG

Connaissance des modes de publication d'une application web

Connaissance des principes de base du référencement

Connaissance des normes ECMAScript et du DOM du W3C

Connaissance des langages et framework de script client

Connaissance des principales failles de sécurité des applications web (XSS, CSRF...) et de leurs parades

Connaissance des bonnes pratiques de sécurisation des clients web

Learning/Training

<https://thinkster.io/tutorials/getting-started-with-react>

<https://www.freecodecamp.org/learn/front-end-libraries/react/create-a-simple-jsx-element>

<https://www.freecodecamp.org/news/react-examples-reactjs/>

<https://openclassrooms.com/fr/courses/4664381-realisez-une-application-web-avec-react-js/4664801-demarrez-facilement-avec-create-react-app>

Youtube :

<https://www.youtube.com/watch?v=no82oluCZag&t=1s>

https://www.youtube.com/watch?v=LuxYWWB3_0c

Hook et HOC

<https://dev.to/gethackteam/from-higher-order-components-hoc-to-react-hooks-2bm9>

<https://www.winzana.com/article/hoc-to-hook>

<https://www.freecodecamp.org/news/react-hooks-to-hocs/>

<https://www.w3resource.com/react/glossary-of-react-terms.php>

<https://medium.com/@onoufriosm/real-world-examples-of-higher-order-components-hoc-for-react-871f0d8b39d8>

<https://medium.com/@onoufriosm/real-world-examples-of-higher-order-components-hoc-for-react-871f0d8b39d8>

<https://www.codingame.com/playgrounds/8595/reactjs-higher-order-components-tutorial>

<https://www.freecodecamp.org/news/react-hooks-to-hocs/>

Tests/StoryBook

<https://blog.usejournal.com/testing-with-jest-and-enzyme-in-react-part-3-best-practices-when-testing-with-jest-and-enzyme-ae3fe0c39d06>

<https://wanago.io/2018/09/17/javascript-testing-tutorial-part-four-mocking-api-calls-and-simulating-react-components-interactions/>

<https://medium.com/@jondot/how-to-test-a-full-react-app-using-nothing-but-storybook-15f4c584e30a>

<https://blog.logrocket.com/react-storybook/>

Communication entre composants

<https://itnext.io/passing-data-between-sibling-components-in-react-using-context-api-and-react-hooks-fce60f12629a>

<https://www.javascriptstuff.com/component-communication/>

<https://hackernoon.com/everything-you-need-to-know-about-reacts-context-api-e5c8c32ef202>

<https://scotch.io/courses/10-react-challenges-beginner/use-context-to-pass-data?autoplay=true>

<https://react-devtools-tutorial.now.sh/element-selector-tool>

Atelier 1 :

Comprendre la philosophie react : Lundi matin

1. Comparer les trois framework : Angular, VueJs et React

<https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>

2. Prise en main de React : RTFM !!! :)

<https://reactjs.org/docs/getting-started.html>

3. Pensez une interface en react

<https://fr.reactjs.org/docs/thinking-in-react.html>

<https://www.innominds.com/blog/how-to-build-your-first-react-js-application>

Entraînez-vous à décomposer les interfaces en composants. Cela vous aidera pour votre chef d'oeuvre

4. Exercices d'initialisation basiques :

https://www.w3schools.com/REACT/react_intro.asp

5. Exercices pour s'initier à React en mode TDD :

<https://github.com/4GeeksAcademy/react-exercises>

6. Communication entre composants

React étant basé sur des composants, il sera nécessaire de connaître les différents pattern et méthodes pour permettre la communication des data entre composants

- . parents - enfants
- . enfants -> parents
- . sibling components

Une problématique : comment gérer le state entre les composants ?

Lors des prochains ateliers, nous verrons plus en détails l'architecture d'une app avec Redux qui est plus adoptée en entreprise, et permet de gérer plus efficacement les states.

Quizz :

Quels sont les trois principes fondamentaux de React?

Quels sont les deux types de composants dans React?

Quels sont les problèmes avec les composants de classe dans React?

Quel outil pouvons-nous utiliser pour configurer un projet React et quelle est la commande que nous devons exécuter pour l'utiliser?

Comment accéder à l'état React et le définir à l'aide de crochets?

Quels sont les avantages de l'utilisation de composants de fonction avec des crochets par rapport aux composants de classe?

Avons-nous besoin de remplacer tous les composants de classe par des composants de fonction utilisant des crochets lors de la mise à jour de React?

Quels sont les trois crochets de base fournis par React?

Mot clés

- jsx
- virtual dom
- Function & Class Components
- Life cycle, state et props et ref
- hooks
- contexte
- router
- HOC
- jest
- les props-types
- Gestion d'un state globale
- Styles avec react
- React DevTools

Atelier 2 :

Créer un mini app avec Swappi :

Lundi après midi, mardi matin, en groupe de 2



INTRODUCTION

Le site <https://swapi.co/> est une grande base de données sur Star Wars.

Il vous met à disposition une API qui vous permet de récupérer des informations rapidement et facilement en format JSON.

Par exemple si nous voulons récupérer la liste de tous les personnages il nous suffit de faire la requête HTTP suivante: <https://swapi.co/api/people/>.

Prends le temps de bien lire la documentation ainsi que les réponses que tu recevras en JSON. Tu devras réaliser une application ReactJS qui proposera une navigation dans cette base de données.

Conseil : Lire les différentes étapes avant de commencer

Démarrer facilement avec CRA (create-react-app) qui contient une configuration de base, notamment pour transpiler le code jsx en ES5

+ ETAPE 1

Faire une page qui affiche tous les personnages de la série.

Ajoute une barre de recherche qui permet de rechercher un personnage par nom.

La recherche doit afficher les résultats en temps réel.

Indice :

- un composant Home avec un state : personnages
- une requête récupérant les personnages à l'initialisation du composant Home en utilisant les life cycles
- le résultat de la requêtes change le state personnages
- un input et une fonction onChange qui fera une requête recherchant les personnages avec la valeur de l'input, puis changera le state personnages avec les résultats
- on peut imaginer faire un composants Personnages prenant en props les personnages

+ ETAPE 2

Faire en sorte que lorsqu'on clique sur un personnage, on arrive sur sa page avec les informations suivantes: le nom, la couleur des yeux, l'année de naissance, le genre, le nom des vaisseaux spatiaux piloté (starships), la date de création et d'édition de la ressource.

Indice :

- react-router
 - . Component Home à l'url '/'
 - . Component Personnage à l'url '/personnages/:id'
- routes dynamique -> /personnages/:id
- vous pouvez récupérer le numéro id de l'url de l'API pour le passer en paramètre du <Link> réfèrent à la route.
- Dans le composant Personnage, vous pourrez récupérer l'id de l'url (match) pour pouvoir requêter les informations le concernant.

+ ETAPE 3

Sur la page d'un personnage, il doit être possible de cliquer sur un nom de vaisseau spatial piloté par le personnage, et d'être redirigé sur la page d'information du vaisseau spécifique.

Les informations suivantes devront être affichées: le nom du vaisseau, le modèle, le fabricant et les différentes personnes ayant déjà piloté ce vaisseau spatial.

Il doit être possible de cliquer sur un pilote afin d'être redirigé vers sa page d'information que vous avez faite dans l'étape précédente

Indice :

- routes dynamiques
- Page informations avec props

+ ETAPE 4 (Bonus) Libre à toi de tester l'application à l'aide de Jest et Enzyme, si tu as le temps bien sûr !



+ ETAPE 5 Effectuer une code reviews du code d'un autre groupe

Atelier 3 :

Créer votre page perso avec React



<https://www.welcometothejungle.com/fr/articles/book-portfolio-avantages>

Consigne :

Le but de ce projet est de constituer la base de votre mini-book/portfolio.

Vous pourrez mettre en place json-server pour simuler une base de donnée de projets (google est ton ami)

Vous pouvez par exemple :

Créer différentes pages Components

- Une page AboutMe contenant vos compétences-skills et vos technologies favorites
- Une page Projets pour présenter vos beaux projets
- Une page Contact

Utiliser le react-router, pour permettre la navigation entre les pages.

Faite rapidement un zoning de votre portfolio, et imaginez les différents composants à créer

Bonus : faite fonctionner le formulaire coté back-end (nodemailer),

Faites des tests avec Jest et enzyme

Exemples :

<https://www.barrymichaeldoyle.com/#/contact>

Atelier 4 :

Hébergement avec Heroku



Il est temps de concrétiser votre développement et de mettre en ligne votre app.
Votre mission sera d'héberger votre book avec Heroku

Indice :

- Créer un server node basique
- Builder votre app react, transpilation
- .env : Créer une constante URL pour les différent environnement (dev, prod)

<https://medium.com/jeremy-gottfrieds-tech-blog/tutorial-how-to-deploy-a-production-react-app-to-heroku-c4831dfcfa08>

<https://www.freecodecamp.org/news/how-to-deploy-a-react-app-with-an-express-server-on-heroku-32244fe5a250/>