|  |
| --- |
| 420-KVA-JQ Web transactionnel |

Dans ce laboratoire, vous aurez à modifier le site Web du laboratoire #2 afin d’utiliser React.

# Laboratoire #3



Votre tâche est d’utiliser React afin d’améliorer l’organisation de votre projet en le découpant en composants.

Contexte

Le client est très satisfait du blog que vous lui avez fourni à l’itération précédente. Votre chef d’équipe, un peu moins.

Il n’a pas de problème avec l’implémentation actuelle, mais il a un peu d’expérience en gestion de projet et sait que l’application risque d’être difficile à faire évoluer et à maintenir si l’équipe n’essaie pas tout de suite de mieux en gérer la complexité.

Après une analyse exhaustive de l’application et une veille technologique afin de connaître les bonnes pratiques récentes en développement Web, il est déterminé qu’une refonte de l’application en utilisant un *framework* JavaScript moderne s’impose. L’entreprise décide d’y aller pour le choix sûr en utilisant le *leader* de l’industrie : React.

Ajout des librairies

Ajoutez les librairies permettant d’utiliser React et Babel (pour utiliser JSX) dans vos fichiers HTML. Afin de l’intégrer progressivement, intégrez-les en utilisant les CDN.

**Attention** : Assurez-vous que les librairies sont ajoutées **avant** vos scripts et dans l’ordre suivant :

1. React
2. React-dom
3. Babel

Assurez-vous également d’inclure les scripts contenant du JSX en précisant le type « text/babel ». De plus, assurez-vous que les scripts de vos composants soient inclus avant vos scripts qui les utilisent!

Création de la racine de React

Dans chacune des pages HTML, ajoutez une balise DIV servant de racine pour React (root) et un script contenant le code permettant de l’initialiser.

Comme l’intégration de React se fera progressivement, il est recommandé de ne rien supprimer dans le HTML pour l’instant.

Testez l’intégration de React en essayant d’injecter une balise <p> contenant « Hello world » dans la racine.

Composant Header

Créez un composant « Header » qui servira d’entête à vos pages.

Copiez-y le code de votre entête et injectez le composant dans la page.

Supprimez le code de l’entête dans le HTML.

Composant Footer

Créez un composant « Footer » qui servira de pied de pages.

Copiez-y le code de votre pied de page et injectez le composant dans la page.

Supprimez le code du pied de page dans le HTML.

Composant BlogCard

Créez un composant « BlogCard » qui représentera une carte de la page principale de votre blog.

Le composant doit pouvoir être paramétré en utilisant les props.

Testez le composant en l’injectant dans la page.

Composant BlogList

Créez un composant « BlogList » qui contiendra la liste des blogs de la page principale.

Ce composant sera responsable de charger les blogs à partir de l’API (en utilisant useEffects). Il devra également générer les cartes à partir de la liste récupérée.

Quand le composant sera fonctionnel, injectez-le et supprimez le reste du code HTML de la page principale.

Composant BlogDetails

Créez un composant « BlogDetails » qui servira à afficher les détails dans la page de blog.

Le composant doit recevoir le ID de l’article de blog grâce aux props.

Ce composant sera responsable de charger les détails de l’article à partir de l’API (en utilisant useEffects).

Testez le composant en l’injectant dans la page.

Composant Comment

Créez un composant « Comment » qui servira à afficher un commentaire dans la page de blog.

Le composant doit pouvoir être paramétré en utilisant les props.

Testez le composant en l’injectant dans la page.

Composant CommentList

Créez un composant « CommentList » qui servira à afficher la liste des commentaires dans la page de blog.

Le composant doit recevoir le ID de l’article de blog grâce aux props.

Ce composant sera responsable de charger les commentaires à partir de l’API (en utilisant useEffects). Il devra également générer les commentaires à partir de la liste récupérée.

Testez le composant en l’injectant dans la page.

Composant AddComment

Créez un composant « AddComment » qui servira à afficher la zone permettant de créer un nouveau commentaire.

Le composant doit recevoir le ID de l’article de blog grâce aux props.

Ce composant sera responsable d’envoyer le commentaire à l’API.

Testez le composant en l’injectant dans la page.

Composant Blog

Créez un composant « Blog » qui servira de conteneur pour la page de blog.

Ce composant doit récupérer l’ID du blog à partir de l’URL et afficher les détails de l’article, ainsi que la section des commentaires.

Injectez-le dans la page et supprimez ce qui reste de code HTML.

Grille de correction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Excellent**  **(5)** | **Très bien**  **(4)** | **Bien**  **(3)** | **Suffisant**  **(2)** | **Insuffisant**  **(0-1)** |
| **Installation de React** |  |  |  |  |  |
| **Composant Header** |  |  |  |  |  |
| **Composant Footer** |  |  |  |  |  |
| **Composant BlogCard** |  |  |  |  |  |
| **Composant BlogList** |  |  |  |  |  |
| **Composant BlogDetails** |  |  |  |  |  |
| **Composant Comment** |  |  |  |  |  |
| **Composant CommentList** |  |  |  |  |  |
| **Composant AddComment** |  |  |  |  |  |
| **Composant Blog** |  |  |  |  |  |
| **Organisation générale du code** |  |  |  |  |  |

Total : / 55

Remise

Vous devez remettre votre application sur Git. La date de remise est indiquée sur Teams.