

## TP 2 – Blog d'experts

Maintenant que vous avez réussi à mettre en ligne les recettes et permis à vos utilisateurs de les consulter intégralement ou bien en filtrant leurs noms, vous allez rajouter une nouvelle fonctionnalité à votre site.

Vous allez créer un petit blog afin que vos utilisateurs (chefs cuisiniers et experts en cuisine) puissent s'exprimer et poster des articles autour de leurs spécialités culinaires.

Dans ce Tp, vous allez ainsi apprendre à lire des données à partir d'un serveur json, faire des requêtes CRUD à partir d'une mini base de données NOSQL.

### Menu et routage

1. Tout d'abord, créer le composant "Recettes" qui affichera vos données au lieu de les afficher directement dans "App".
2. Créer le composant "Blog" qui contiendra le formulaire de saisie ainsi que tous les articles déjà postés.
3. Créer le composant "Article" qui contiendra les données d'un article posté.
4. Créer un composant menu avec un onglet "Recettes" pour afficher les recettes et un onglet "Blog" pour afficher le formulaire et les articles dans une page différente. Importer et utiliser le composant "NavLink" (<https://reactrouter.com/en/main/components/nav-link>).
5. Dans "App.js", importer et utiliser "BrowserRouter" et les composants associés afin de gérer des routes différentes qui correspondent aux deux onglets de votre menu ([https://www.w3schools.com/react/react\\_router.asp](https://www.w3schools.com/react/react_router.asp)).
6. Le menu doit être présent sur les deux pages : Recettes et Blog avec une mise en valeur de l'onglet actif (voir image ci-dessous).

### Spécification pour le composant Blog

#### Affichage des articles

- Votre blog doit afficher les articles existant. Vous utiliserez ainsi le fichier "db.json" fourni sur Chamilo.
- Dans "src", créer un répertoire "assets" et y importer le fichier "db.json".
- Installer JSON-SERVER à travers la commande : **npm i -g json-server**
- Exécuter directement "**json-server --w src/assets/db.json --port 3003**" pour faire tourner votre serveur (si le port 3003 est occupé, vous pouvez utiliser un autre port ; 3004 par exemple).
- Optionnellement, vous pouvez importer la commande ci-dessus dans les "scripts" de votre fichier "package.json" de cette façon. Ceci vous permettra de lancer le serveur directement à travers la commande "**npm run server**"

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject",  
  "server": "json-server --w src/assets/db.json --port 3003"  
},
```

- Utiliser axios pour lire vos données directement depuis le json. Le principe est le même que pour les recettes sauf que l'url sera la suivante :  
**"http://localhost:3003/articles"**

## Formulaire de saisie

- Utiliser des variables d'état pour enregistrer l'auteur et l'article.
- Mettre un contrôle de saisie afin de ne pouvoir pas soumettre un message dont la taille est inférieure à 100 caractères (penser à utiliser une variable d'état également).
- La soumission des articles se fait également avec axios (méthode post). Il faudra respecter le format des données dans le json (author, content, date).

The screenshot displays a web interface for a blog. At the top, there are two tabs: 'Recettes' and 'Blog', with 'Blog' being the active tab. Below the tabs is a large heading 'Blog'. Underneath, there is a form with two input fields: 'Nom' and 'Message'. The 'Message' field has a red border and a red error message below it: 'Veuillez écrire un minimum de 140 caractères'. To the right of the 'Message' field is a red 'Envoyer' button. Below the form, there are three blog posts displayed in a list. Each post has a title, a date, a content snippet, and two buttons: 'Modifier' and 'Supprimer'.

Recettes Blog

## Blog

Nom

Message

Veuillez écrire un minimum de 140 caractères

Envoyer

**Master Chef** Posté le 21 novembre 2022 à 20:30:12

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente doloribus vel sunt laborum eos enim odit maxime corporis vero nihil voluptatum a, ullam delectus ad sit. Odit quis adipisci illo sit porro recusandae? Sequi, dolorum repellendus doloremque qui tempora beatae.

Modifier Supprimer

**Philippe Etchebest** Posté le 21 novembre 2022 à 20:29:50

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente doloribus vel sunt laborum eos enim odit maxime corporis vero nihil voluptatum a, ullam delectus ad sit. Odit quis adipisci illo sit porro recusandae? Sequi, dolorum repellendus doloremque qui tempora beatae. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente doloribus vel sunt laborum eos enim odit maxime corporis vero nihil voluptatum a, ullam delectus ad sit. Odit quis adipisci illo sit porro recusandae? Sequi, dolorum repellendus doloremque qui tempora beatae.

Modifier Supprimer

**Gordon Ramsay** Posté le 21 novembre 2022 à 20:29:24

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente doloribus vel sunt laborum eos enim odit maxime corporis vero nihil voluptatum a, ullam delectus ad sit. Odit quis adipisci illo sit porro recusandae? Sequi, dolorum repellendus doloremque qui tempora beatae. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente doloribus vel sunt laborum eos enim odit maxime corporis vero nihil voluptatum a, ullam delectus ad sit.

Modifier Supprimer

Exemple d'affichage attendu du formulaire et des articles postés

## Spécification pour le composant Article

- Affiche les données d'un article, à savoir l'auteur, la date et le contenu.
- Permet de supprimer un article.
  - Il est recommandé d'utiliser une variable d'état pour indiquer si un article est en mode édition (et donc afficher l'input) ou non.
- Toute suppression ou édition d'un article doit mettre à jour l'affichage sans recharger la page.