Tables des matières

[Technologies utilisés 2](#_Toc102692986)

[Front-end 2](#_Toc102692987)

[Back-end 2](#_Toc102692988)

[Données 2](#_Toc102692989)

[Hébergement 2](#_Toc102692990)

[Notes importantes 2](#_Toc102692991)

[Pages de l’application 2](#_Toc102692992)

[Accueil 2](#_Toc102692993)

[Liste 2](#_Toc102692994)

[Créer (formulaire) 2](#_Toc102692995)

[Créer événement automatiquement via Backend REST API 3](#_Toc102692996)

[Tests unitaires 5](#_Toc102692997)

[Déploiement Site web 5](#_Toc102692998)

[Github 5](#_Toc102692999)

# Technologies utilisés

## Front-end

|  |  |
| --- | --- |
| **Technologie** | **Expérience** |
| React | 0 |
| Javascript | ~3 ans |
| HTML | ~5 ans |
| CSS | ~5 ans |

## Back-end

|  |  |
| --- | --- |
| **Technologie** | **Expérience** |
| Nodejs | 0 |
| Express | 0 |

## Données

|  |  |
| --- | --- |
| **Technologie** | **Expérience** |
| JSON | Un peu |

## Hébergement

|  |  |
| --- | --- |
| **Technologie** | **Expérience** |
| GIT | ~3 ans |
| Heroku | 0 |

# Notes importantes

## Pages de l’application

### Accueil

Page d’accueil normale, rien de spécial ici.

### Liste

Cette page affiche la liste entière des événements présent dans le fichier evenements.JSON présent dans le backend.

Il y a aussi un formulaire de recherche simple, pour cibler des événements en particulier.

### Créer (formulaire)

Cette page contient un formulaire pour créer un nouvel événement. **IMPORTANT!** L’envoi des données du formulaire vers le Backend n’est pas encore implémenté, par faute de temps et de manque de connaissance technique.

La liste des fuseaux horaires provient de <https://github.com/dmfilipenko/timezones.json>.

Les boutons radios « Show Daylight Saving Times » et « Show Standard Times », une fois sélectionnés, cacheront les Fuseaux Horaires non correspondants de la liste des fuseaux horaires. Le fuseau horaire présentement choisi ne changera pas.

Une fois la date, l’heure et le fuseau horaire choisi, l’heure de l’événement selon le fuseau horaire de Québec sera alors affichée.

### Créer événement automatiquement via Backend REST API

Cette page avait pour but de créer un événement automatiquement via le back end afin de prouver que je puisse ajouter des éléments à un fichier JSON, mais pour une raison obscure la fonction du Back-End que j’ai codé pour ajouter un événement est appelé à chaque fois que je lance le serveur, où que je rafraichis forcément une page (Ctrl + f5), créant ainsi plusieurs copies du même événement.  
  
Dans mon App.js (Front-End)

function CreerEvenement()

{

  const [data, setData] = React.useState(null);

  React.useEffect(() => {

    fetch("/api/creerEvenement")

      .then((res) => res.json())

      .then((data) => setData(data.message));

  }, []);

  return (<div className='creerEvenement'>

  <main>

    <h2>Page débug - Creer un Evenement</h2>

    <p>Cette page appelle un REST API qui ajoute un événement.</p>

    <p>Malheureusement, je n'ai pas eu le temps pour soumettre des données au REST API via un formulaire en ReactNodeJS</p>

    <p>Mais au moins, je peux créer une nouvelle entrée dans un fichier JSON à partir du backend.</p>

    <p>{!data ? "Loading..." : data}</p>

    <p><NavLink exact activeClassName="current" to='/liste'>Liste événements</NavLink></p>

    <div id="prefooter"></div>

  </main>

</div>);

}

Dans mon Index.js (Back-End)

app.get("/api/creerEvenement", (req,res) =>

{

  //Append

  const newEvenement =

  {

    nom: "Événement créé automatiquement via debug",

    date: "2022/07/07",

    heure: "20:00",

    timezone: "Eastern Standard Time",

    offset: -5,

    description: "Insert description here",

    siteWeb: "www.google.ca"

  }

  res.json({ message: "Test" });

  //Mettre à true pour effectuer l'ajout d'un événement

  if (false)

  {

    jsonReader("./server/evenements.JSON", (err, evenements) =>

    {

      if (err)

      {

        console.log(err);

        return;

      }

      try

      {

        evenements.push(newEvenement);

        fs.writeFile("./server/evenements.JSON", JSON.stringify(evenements, null, 2), function(err)

        {

          if (err) throw err;

          console.log('The "data to append" was appended to file!');

        });

      }

      catch (err)

      {

        console.log("Error parsing JSON", err);

      }

    })

  }

});

## Tests unitaires

Non implémenté par manque de temps. Bien que j’aurais voulu commencer par les tests unitaires afin d’avoir une méthodologie Test Driven Development, j’ai du les remettre à plus tard à cause de mon manque total d’expérience en React/Node (Je ne sais pas quels sont les tests que je peux créer à l’avance si je ne sais pas comment les fonctions sont structurées!)

## Déploiement Site web

<https://kumojin-application.herokuapp.com/>

## Github

<https://github.com/EtienneChouinard/kumojin-application>

## Lancer application en local

Dans un terminal, dans le dossier \react-node-app :

Npm start

Dans un autre terminal, dans le dossier \react-node-app\client :

Npm Start

Dans un navigateur Web, aller dans http://localhost:3000/