

## Atelier 12

### Objectif d'apprentissage :

- React Router

### Exercice 1 :

1. Créez un nouveau projet React en utilisant create-react-app,
2. Installez React Router dans votre projet
3. Créez deux composants : Home.js et About.js. Ces composants représenteront respectivement la page d'accueil et la page à propos.
4. Configurez React Router dans votre fichier principal (App.js ou index.js).
5. Créez deux routes : une pour la page d'accueil (/) qui affiche le composant Home et une pour la page à propos (/about) qui affiche le composant About.
6. Dans les composants Home.js et About.js, ajoutez du contenu simple pour afficher sur les pages respectives (par exemple, un titre et un paragraphe).
7. Dans le composant Home.js, ajoutez un lien (<Link>) vers la page à propos (/about).
8. Dans le composant About.js, ajoutez un lien de retour à la page d'accueil (/).
9. Ajoutez un peu de style CSS pour rendre vos pages plus attrayantes.
10. Testez votre application en démarrant le serveur de développement (npm start ).

### Exercice 2 :

1. Créez un nouveau projet React en utilisant create-react-app.
2. Installez React Router dans votre projet en utilisant npm
3. Créez deux composants : Home.js et Article.js. Home.js représentera la page d'accueil avec la liste des articles, et Article.js affichera les détails d'un article individuel.
4. Créez un fichier articlesData.js contenant un tableau d'objets représentant des articles. Chaque objet doit avoir des propriétés telles que : id, title, description, image, stock.
5. Configurez React Router dans votre fichier principal (App.js ou index.js).
6. Créez deux routes : une pour la page d'accueil (/) qui affiche le composant Home et une pour les articles individuels (/article/:id) qui affiche le composant Article.
7. Dans le composant Home.js, utilisez les données d'articles pour afficher une liste d'articles.
8. Dans le composant Article.js, utilisez les paramètres d'URL pour récupérer l'ID de l'article à afficher. Affichez le titre et le contenu de l'article avec cet ID.
9. Ajoutez un peu de style CSS pour rendre votre application de blog plus attrayante.
10. Testez votre application.

### Exercice 3 :

#### **1. Configuration du projet :**

- Créez un nouveau projet React en utilisant Create React App ou tout autre outil de votre choix.
- Installez react-router-dom dans votre projet

#### **2. Création des composants :**

Le projet sera structuré de la façon suivante :

```
src/
|-- components/
|   |-- Accueil.js
|   |-- Categories.js
|   |-- Technologie.js
|   |-- Voyages.js
|   |-- Cuisine.js
|   |-- Articles.js
|   |-- ArticleDetails.js
|-- App.js
|-- index.js
```

- Créez trois composants : **Accueil**, **Categories**, et **Articles**.
- Créer les composants **Technologie**, **Voyages**, **Cuisine**, qui seront des sous composants de **Caregories**
- Créer un composant **ArticleDetails** qui accepte comme paramètre ID de l'article.

### 3. Configurer les routes :

- Configurez les routes dans votre application en utilisant BrowserRouter et Route du package react-router-dom.
- Créez une route pour la page d'accueil (/) qui affiche le composant Accueil.
- Créez une route pour la page de catégories (/categories) qui affiche le composant Categories.
- À l'intérieur du composant Categories, configurez des routes imbriquées pour chaque catégorie (par exemple, /categories/technologie, /categories/voyages, /categories/cuisine).
- Créez une route pour la page d'articles (/articles/:id) qui affiche le composant ArticleDetails. Utilisez les paramètres d'URL pour passer l'identifiant de l'article.

### 4. Créer le contenu des composants :

- Dans le composant Accueil, affichez une liste de catégories avec des liens vers les pages de catégories correspondantes.
- Dans chaque sous-composant de Categories, affichez une liste d'articles associés à cette catégorie.
- Dans le composant ArticleDetails, affichez les détails de l'article basé sur l'identifiant de l'article passé en tant que paramètre d'URL.

### 5. Ajouter la navigation :

- Ajoutez des liens de navigation pour accéder à la page d'accueil, à la page de catégories et aux pages d'articles depuis n'importe quelle partie de l'application.
- À l'intérieur de la page de catégories, ajoutez des liens de navigation pour accéder aux pages d'articles associées à chaque catégorie.

## Exercice 4 :

L'API que nous allons utiliser dans cet exercice REST Countries API (<https://restcountries.com/#endpoints-all>). Les endpoints que nous allons utiliser sont :

- **ENDPOINT 1:** <https://restcountries.com/v3.1/all> : pour retourner la liste de tous les pays dans le monde.

- **ENDPOINT 2:** <https://restcountries.com/v3.1/name/{name}> : pour retourner le détails d'un pays passe en paramètre.

### 1. Configurer le projet React :

- Créez un nouveau projet React en utilisant Create React App.
- Installez react-router-dom et axios

- Configurez les routes pour avoir une page d'accueil qui affiche la liste des pays et une page de détails pour chaque pays.

```
src/  
|-- components/  
|   |-- Accueil.jsx  
|   |-- DetailsCountry.jsx  
|-- App.js  
|-- index.js
```

## 2. Création des composants :

- Créez deux composants : Accueil et DetailsCountry.
- Le composant Accueil affiche la liste des pays sous format Grid (image du drapeau et le nom du pays).
- Le composant DetailsCountry affiche les détails d'un film dont le nom est passé en paramètre, tels que le nom, le nom du continent, l'image du drapeau, nombre de population ...

## 3. Navigation entre les pages :

- Utilisez react-router-dom pour mettre en place la navigation entre la page d'accueil et la page de détails.
- Sur la page d'accueil, créez des liens cliquables pour chaque film qui redirigent l'utilisateur vers la page de détails du film correspondant.