Atelier 3 NextJS : CRUD auteurs

# Variable d’environnement

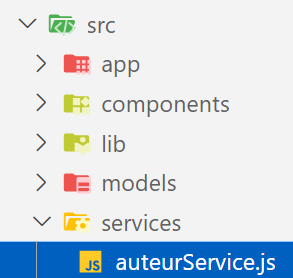


NEXT\_PUBLIC\_API\_URL = http://localhost:3000/api

# Services

Ensuite, sous /src créer le dossier services.

Dans lequel créer auteurService.js



const AUTEUR\_API="/auteurs/"

export const fetchAuteurs=async()=> {

const res = await fetch(process.env.NEXT\_PUBLIC\_API\_URL+AUTEUR\_API, { cache: 'no-store' })

const response = await res.json();

return response;

}

export const fetchAuteurById=async(auteurId)=> {

    const res = await fetch(process.env.NEXT\_PUBLIC\_API\_URL+AUTEUR\_API+`${auteurId}`, { cache: 'no-store' },{

        method: 'GET'

    });

    const response = await res.json();

     return response;

}

export const deleteAuteur=async(auteurId) =>{

    const res = await fetch(process.env.NEXT\_PUBLIC\_API\_URL+AUTEUR\_API+`${auteurId}`,{

        method: 'DELETE'

    });

    const response = await res.json();

    return response;

}

export const addAuteur=async(auteur)=> {

    const res = await fetch(process.env.NEXT\_PUBLIC\_API\_URL+AUTEUR\_API, {

        method: 'POST',

        headers: {

        'Content-Type': 'application/json',

        },

        body: JSON.stringify(auteur),

    });

    const response = await res.json();

    return response;

}

export const editAuteur=async(auteur) =>{

    const res = await fetch(process.env.NEXT\_PUBLIC\_API\_URL+AUTEUR\_API+`${auteur.\_id}`, {

        method: 'PUT',

        headers: {

        'Content-Type': 'application/json',

        },

        body: JSON.stringify(auteur),

    });

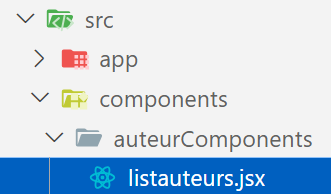
    const response = await res.json();

    return response;

}

# Affichage des auteurs

Sous /src/components/ créer le dossier auteurComponents puis créer le fichier listauteurs.jsx



Accéder au site<https://www.material-react-table.com/>

Puis Quick examples->Basic Example

Installer le composant material-react-table

npm install material-react-table

"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

const Listauteurs = ({auteurs}) => {

  const columns = useMemo(

    () => [

    {

    accessorKey: 'nomauteur', //access nested data with dot notation

    header: 'NOM',

    size: 100,

    },

    {

    accessorKey: 'email',

    header: 'EMAIL',

    size: 100,

    },

    {

    accessorKey: 'numtel', //normal accessorKey

    header: 'TEL',

    size: 100,

    }

    ],

    [auteurs],

    );

return (

<div>

<MaterialReactTable columns={columns} data={auteurs} />;

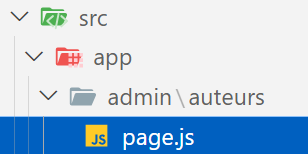
</div>

  )

}

export default Listauteurs

On va créer un dossier appelé auteurs sous /src/app/admin ayant page.js



import Listauteurs from '@/components/auteurComponents/listauteurs';

import {fetchAuteurs} from "@/services/auteurService"

const getAuteurs=async()=>{

const data=await fetchAuteurs()

return data;

}

const AuteurPage = async() =>{

    const auteurs=await getAuteurs()

  return (

   <div className="container">

      <Listauteurs auteurs={auteurs}/>

    </div>

  )

}

export default AuteurPage

Test dans le navigateur

<http://localhost:3000/admin/auteurs>

## 

# File conventions

Remarque : Pour tous ces fichiers de file convention, une fois le fichier créé et resté vide sans code il générera une erreur.

## error.js

Un fichier d'erreurs définit une limite d'erreur d'interface utilisateur pour un segment de route.

Il est utile pour détecter les erreurs inattendues qui se produisent dans les composants serveur et les composants clients et pour afficher une interface utilisateur de secours.

Sous /src/app/admin/auteurs créer le fichier error.js

'use client' // Error components must be Client Components

import { useEffect } from 'react'

export default function Error({ error, reset }) {

  useEffect(() => {

    // Log the error to an error reporting service

    console.error(error)

  }, [error])

  return (

    <div>

      <h2>Something went wrong!</h2>

      <button

        onClick={

          // Attempt to recover by trying to re-render the segment

          () => reset()

        }

      >

        Try again

      </button>

    </div>

  )

}

Un essai est de mettre une URL end point erronée dans le service dans la requête get puis d’essayer l’url : <http://localhost:3000/admin/auteurs>

On aura une page personnalisée d’erreur.

## 

## loading.js

Par défaut, ce fichier est un composant serveur - mais peut également être utilisé comme composant client via la directive "use client".

Sous /src/app/admin/auteurs créer le fichier loading.js

export default function Loading() {

    // Or a custom loading skeleton component

    return <p>Loading...</p>

  }

Loading n'accepte aucun paramètre.

Comme essai le temps de charger la page on peut voir dans l’url : <http://localhost:3000/admin/auteurs>

## 

## not-found.js

Le fichier not-found est utilisé pour restituer l'interface utilisateur lorsque la fonction notFound est lancée dans un segment de route. En plus de proposer une interface utilisateur personnalisée, Next.js renverra un code d'état HTTP 200 pour les réponses diffusées en continu et 404 pour les réponses non diffusées.

En plus de détecter les erreurs notFound() attendues, le fichier racine app/not-found.js gère également toutes les URL sans correspondance pour l'ensemble de votre application. Cela signifie que les utilisateurs qui visitent une URL qui n'est pas gérée par votre application verront l'interface utilisateur exportée par le fichier app/not-found.js.

Sous **/src/app/** créer le fichier not-found.js



import Link from 'next/link'

export default function NotFound() {

  return (

    <div>

      <h2>Not Found</h2>

      <p>Could not find requested resource</p>

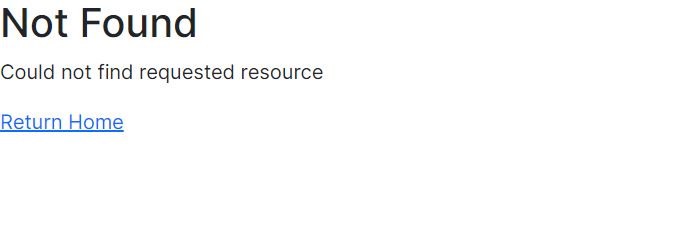
      <Link href="/">Return Home</Link>

    </div>

  )

}

Si on essaie n’importe quelle URL inexistante on aura :



# Suppression des auteurs

On va ajouter l’action de suppression, modifier le contenu de listauteurs.jsx



"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import Button from 'react-bootstrap/Button';

import {deleteAuteur} from "@/services/auteurService"

import { useRouter } from "next/navigation";

import DeleteForeverIcon from '@mui/icons-material/DeleteForever';

const Listauteurs = ({auteurs}) => {

  const router = useRouter();

    const deleteauteur=(id)=>{

      if(window.confirm("supprimer Auteur O/N")) {

        deleteAuteur(id)

        .then((res)=>{ console.log(res)

         router.refresh()

        })

        .catch(error=>{

            console.log(error)

           })

  }

        }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'nomauteur',

    header: 'NOM',

    size: 100,

    },

    {

    accessorKey: 'email',

    header: 'EMAIL',

    size: 100,

    },

    {

    accessorKey: 'numtel',

    header: 'TEL',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

      onClick={(e) => {

      deleteauteur(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <DeleteForeverIcon />

      </Button>

      </div>

      ),

      },

    ],

    [auteurs],

    );

return (

<div>

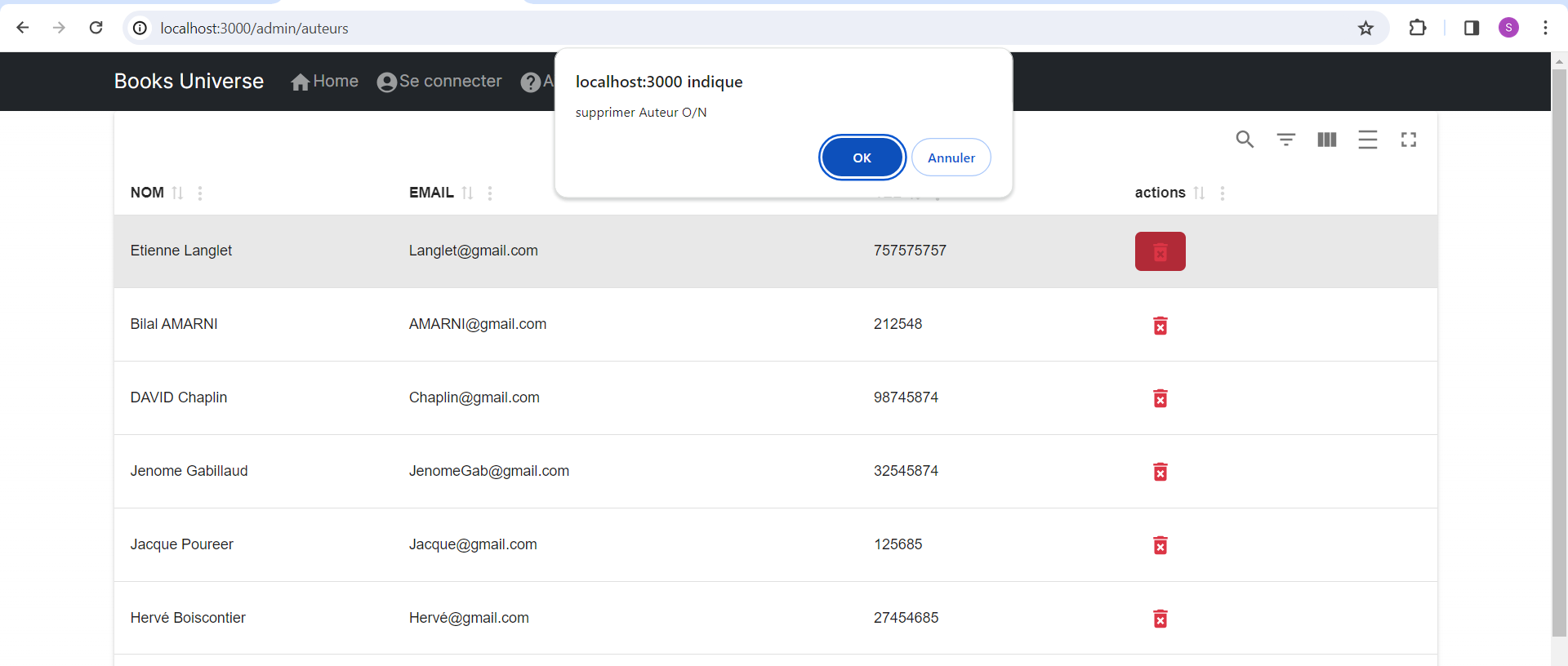
<MaterialReactTable columns={columns} data={auteurs} />

</div>

  )

}

export default Listauteurs



# Ajout d’auteurs

On va ajouter le formulaire d’ajout.

Tout d’abord modifier listauteurs.jsx pour ajouter le bouton de lien vers le formulaire.



"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import Button from 'react-bootstrap/Button';

import {deleteAuteur} from "@/services/auteurService"

import { useRouter } from "next/navigation";

import DeleteForeverIcon from '@mui/icons-material/DeleteForever';

import Link from 'next/link';

import AddCircleOutlineIcon from '@mui/icons-material/AddCircleOutline';

const Listauteurs = ({auteurs}) => {

  const router = useRouter();

    const deleteauteur=(id)=>{

      if(window.confirm("supprimer Auteur O/N")) {

        deleteAuteur(id)

        .then((res)=>{ console.log(res)

         router.refresh()

        })

        .catch(error=>{

            console.log(error)

           })

  }

        }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'nomauteur',

    header: 'NOM',

    size: 100,

    },

    {

    accessorKey: 'email',

    header: 'EMAIL',

    size: 100,

    },

    {

    accessorKey: 'numtel',

    header: 'TEL',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

      onClick={(e) => {

      deleteauteur(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <DeleteForeverIcon />

      </Button>

      </div>

      ),

      },

    ],

    [auteurs],

    );

return (

<div>

<Button

variant='dark'

size="sm"

>

<Link

href="/admin/auteurs/newAuteur"

style={{

  textDecoration: 'none',

  color: 'aqua',

  fontSize: 14,

}}

>

<AddCircleOutlineIcon/> Nouveau

</Link>

</Button>

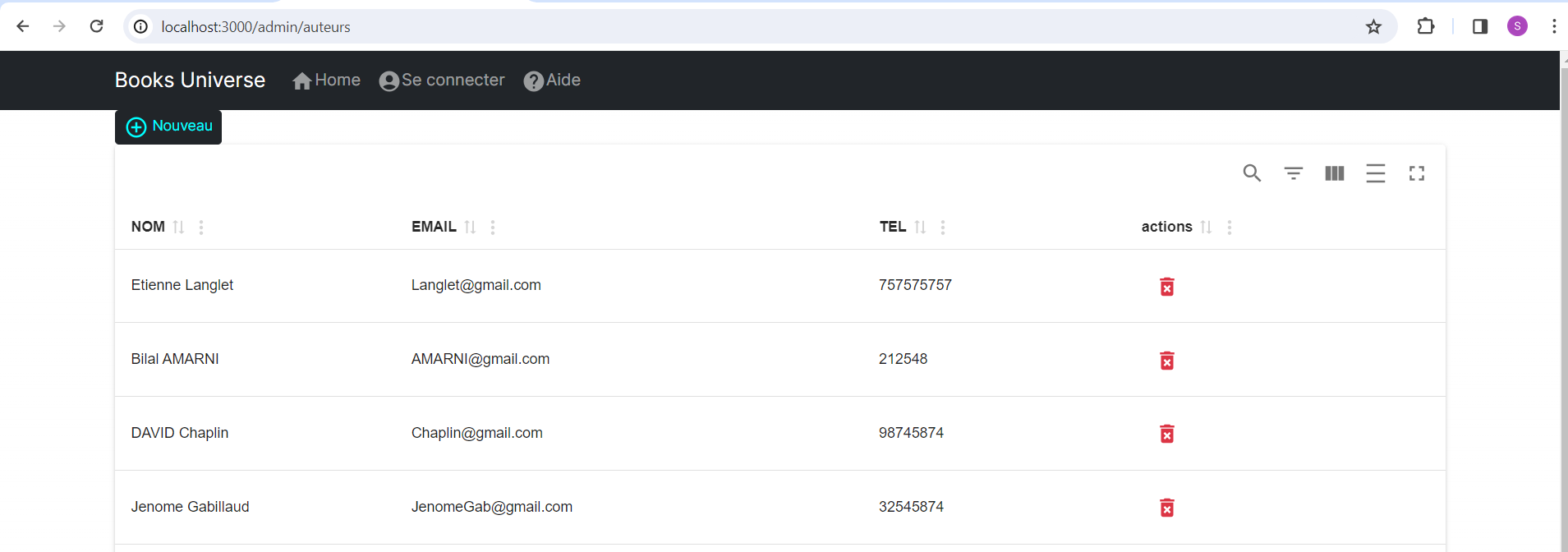
<MaterialReactTable columns={columns} data={auteurs} />

</div>

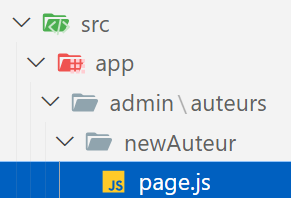
  )

}

export default Listauteurs



Créer sous /src/app/admin/auteurs/ le dossier newAuteur incluant page.js



import NewAuteur from "@/components/auteurComponents/newAuteurComponent";

const NewAuteurPage = async() => {

  return (

    <div>

     <NewAuteur/>

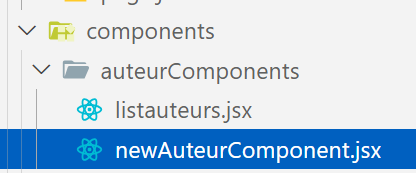
    </div>

  )

}

export default NewAuteurPage

Par la suite, créer sous /src/components/auteurComponents le fichier newAuteurComponent.jsx



"use client"

import React, { useState } from 'react';

import Button from 'react-bootstrap/Button';

import Form from 'react-bootstrap/Form';

import Col from 'react-bootstrap/Col';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import { useRouter } from "next/navigation";

import { addAuteur } from '@/services/auteurService';

const NewAuteur = () => {

  const router = useRouter();

  const [nomauteur, setNomauteur] = useState("");

const [email, setEmail] = useState("");

const [numtel, setNumtel] = useState("");

const [validated, setValidated] = useState(false);

const handleSubmit = (e) => {

e.preventDefault();

const form = e.currentTarget;

if (form.checkValidity() === true) {

const newAuteur = {

    nomauteur,

    email,

    numtel

};

//faire le add dans la BD

addAuteur(newAuteur)

.then(res => {

  router.push('/admin/auteurs')

    router.refresh()

})

.catch(error=>{

alert("Erreur ! Insertion non effectuée")

})

}

setValidated(true);

}

const handleReset = () => {

    setNomauteur("")

    setEmail("")

    setNumtel("")

}

return (

<div>

 <Form noValidate validated={validated} onSubmit={handleSubmit}>

 <h2>Ajout Auteur</h2>

<div className="container w-100 d-flex justify-content-center">

<div>

<div className='form mt-3'>

<Row className="mb-2">

<Form.Group as={Col} md="8" >

<Form.Label >Nom Auteur \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Nom Auteur"

value={nomauteur}

onChange={(e)=>setNomauteur(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Nom Auteur

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="8">

<Form.Label>Email \*</Form.Label>

<Form.Control

required

type="email"

placeholder="Email"

value={email}

onChange={(e)=>setEmail(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Email

</Form.Control.Feedback>

</Form.Group>

</Row>

<Row className="mb-2">

<Form.Group className="col-md-8">

<Form.Label>Téléphone \*</Form.Label>

<InputGroup hasValidation>

<Form.Control

type="number"

required

placeholder="Téléphone"

value={numtel}

onChange={(e)=>setNumtel(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Téléphone Incorrecte

</Form.Control.Feedback>

</InputGroup>

</Form.Group>

</Row>

</div>

</div>

</div>

<center>

<Button type="submit">Enregistrer</Button>

<Button type="button" className="btn btn-warning"

onClick={()=>handleReset()}>Annuler</Button>

</center>

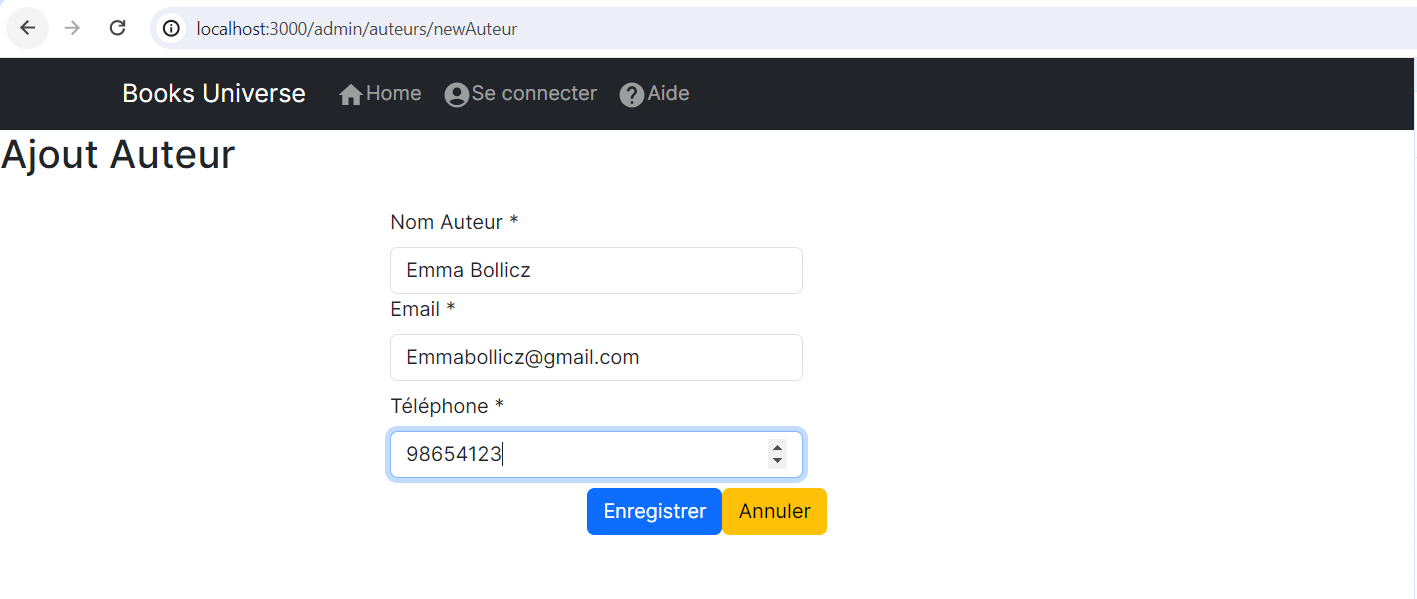
</Form>

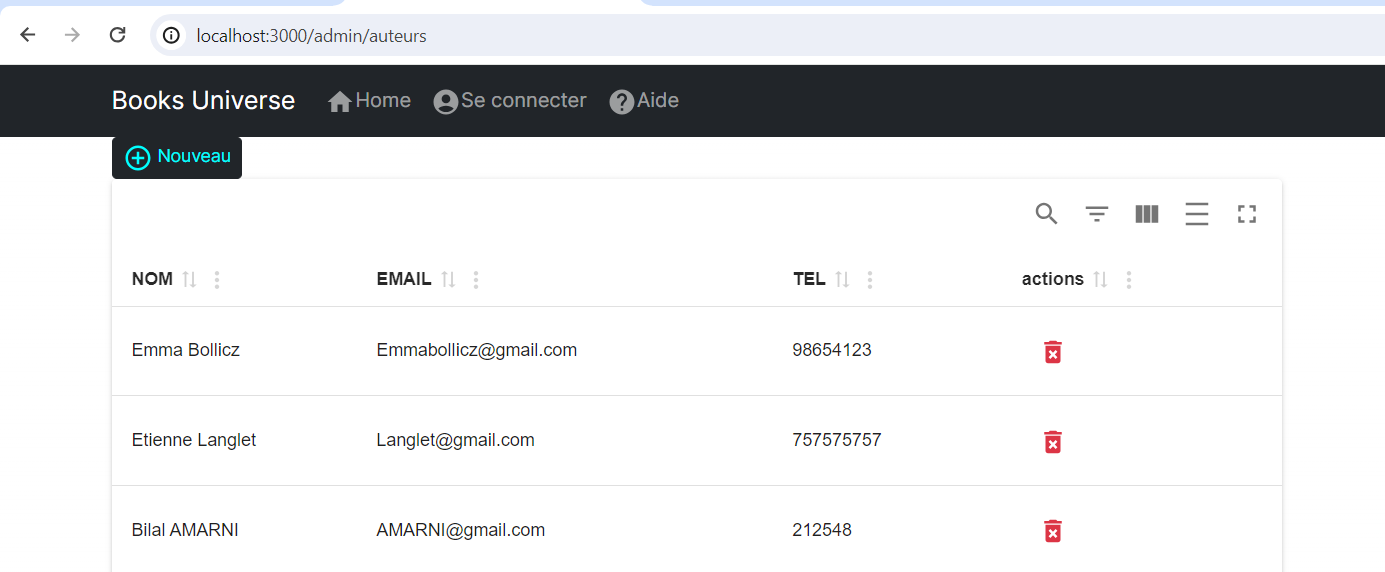
</div>

);

};

export default NewAuteur





# Modification des auteurs

Maintenant, on va procéder à la création du formulaire de la modification.

D’abord, modifier listauteurs pour ajouter le bouton Link vers la page edit.



"use client"

import  { useMemo } from 'react';

import { MaterialReactTable } from 'material-react-table';

import Button from 'react-bootstrap/Button';

import {deleteAuteur} from "@/services/auteurService"

import { useRouter } from "next/navigation";

import DeleteForeverIcon from '@mui/icons-material/DeleteForever';

import Link from 'next/link';

import AddCircleOutlineIcon from '@mui/icons-material/AddCircleOutline';

import EditOutlinedIcon from '@mui/icons-material/EditOutlined';

const Listauteurs = ({auteurs}) => {

  const router = useRouter();

    const deleteauteur=(id)=>{

      if(window.confirm("supprimer Auteur O/N")) {

        deleteAuteur(id)

        .then((res)=>{ console.log(res)

         router.refresh()

        })

        .catch(error=>{

            console.log(error)

           })

  }

        }

  const columns = useMemo(

    () => [

    {

    accessorKey: 'nomauteur',

    header: 'NOM',

    size: 100,

    },

    {

    accessorKey: 'email',

    header: 'EMAIL',

    size: 100,

    },

    {

    accessorKey: 'numtel',

    header: 'TEL',

    size: 100,

    },

    {

      accessorKey: '\_id',

      header: 'actions',

      size: 100,

      Cell: ({ cell, row }) => (

      <div >

      <Button

       size="md"

       className="text-primary btn-link edit"

      >

        <Link href={`/admin/auteurs/updateAuteur/${cell.row.original.\_id}`}>

        <EditOutlinedIcon/>

    </Link>

      </Button>

      <Button

      onClick={(e) => {

      deleteauteur(cell.row.original.\_id,e);

      }}

      variant="danger"

      size="md"

      className="text-danger btn-link delete"

      >

      <DeleteForeverIcon />

      </Button>

      </div>

      ),

      },

    ],

    [auteurs],

    );

return (

<div>

<Button

variant='dark'

size="sm"

>

<Link

href="/admin/auteurs/newAuteur"

style={{

  textDecoration: 'none',

  color: 'aqua',

  fontSize: 14,

}}

>

<AddCircleOutlineIcon/> Nouveau

</Link>

</Button>

<MaterialReactTable columns={columns} data={auteurs} />

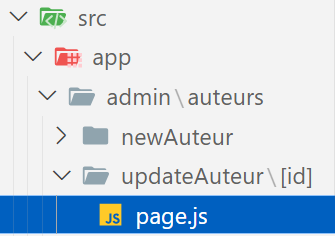
</div>

  )

}

export default Listauteurs

Ensuite, sous /src/app/admin/auteurs créer le dossier updateAuteur puis le dossier **[id]** dans lequel créer page.js



import {fetchAuteurById} from "@/services/auteurService"

import UpdateAuteur from '@/components/auteurComponents/updateAuteurComponent';

const getauteur=async(id)=>{

  const data=await fetchAuteurById(id)

  return data;

}

const AuteurUpdatePage = async({params}) => {

  const auteur = await getauteur(params.id)

  return (

    <div>

       <UpdateAuteur auteur={auteur}  />

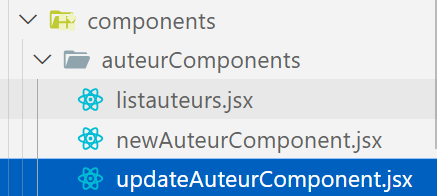
    </div>

  )

}

export default AuteurUpdatePage

Ensuite, créer le composant updateAuteurComponent.jsx sous /src/components/auteurComponents



"use client"

import React, { useState,useEffect } from 'react';

import Button from 'react-bootstrap/Button';

import Form from 'react-bootstrap/Form';

import Col from 'react-bootstrap/Col';

import InputGroup from 'react-bootstrap/InputGroup';

import Row from 'react-bootstrap/Row';

import { useRouter } from "next/navigation";

import { editAuteur } from '@/services/auteurService';

const UpdateAuteur = ({auteur}) => {

  const router = useRouter();

const [nomauteur, setNomauteur] = useState("");

const [email, setEmail] = useState("");

const [numtel, setNumtel] = useState("");

const [validated, setValidated] = useState(false);

useEffect(() => {

    setNomauteur(auteur.nomauteur)

    setEmail(auteur.email)

    setNumtel(auteur.numtel)

 }, [auteur]);

const handleSubmit = (e) => {

e.preventDefault();

const form = e.currentTarget;

if (form.checkValidity() === true) {

const newAuteur = {

    \_id:auteur.\_id,

    nomauteur,

    email,

    numtel

};

//faire le add dans la BD

editAuteur(newAuteur)

.then(res => {

  router.push('/admin/auteurs')

    router.refresh()

})

.catch(error=>{

alert("Erreur ! Insertion non effectuée")

})

}

setValidated(true);

}

const handleReset = () => {

    setNomauteur("")

    setEmail("")

    setNumtel("")

}

return (

<div>

 <Form noValidate validated={validated} onSubmit={handleSubmit}>

 <h2>Modification Auteur</h2>

<div className="container w-100 d-flex justify-content-center">

<div>

<div className='form mt-3'>

<Row className="mb-2">

<Form.Group as={Col} md="8" >

<Form.Label >Nom Auteur \*</Form.Label>

<Form.Control

required

type="text"

placeholder="Nom Auteur"

value={nomauteur}

onChange={(e)=>setNomauteur(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Nom Auteur

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="8">

<Form.Label>Email \*</Form.Label>

<Form.Control

required

type="email"

placeholder="Email"

value={email}

onChange={(e)=>setEmail(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Saisir Email

</Form.Control.Feedback>

</Form.Group>

</Row>

<Row className="mb-2">

<Form.Group className="col-md-8">

<Form.Label>Téléphone \*</Form.Label>

<InputGroup hasValidation>

<Form.Control

type="number"

required

placeholder="Téléphone"

value={numtel}

onChange={(e)=>setNumtel(e.target.value)}

/>

<Form.Control.Feedback type="invalid">

Téléphone Incorrecte

</Form.Control.Feedback>

</InputGroup>

</Form.Group>

</Row>

</div>

</div>

</div>

<center>

<Button type="submit">Enregistrer</Button>

<Button type="button" className="btn btn-warning"

onClick={()=>handleReset()}>Annuler</Button>

</center>

</Form>

</div>

);

};

export default UpdateAuteur

