Atelier 8 NextJS : Server Actions

Les Server Actions sont des fonctions asynchrones exécutées sur le serveur. Ils peuvent être utilisés dans les composants serveur et client pour gérer les soumissions de formulaires et les mutations de données dans les applications Next.js.

# Dépendance

Installer le package suivant :

npm i react-hot-toast

# Service

Modifier le service specialiteService.js sous le dossier /src/services



const SPECIALITE\_API="/specialites/"

export const fetchSpecialites=async()=> {

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

const response = await res.json();

return response;

}

export const fetchSpecialiteById=async(specialiteId)=> {

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

        method: 'GET'

    });

    const response = await res.json();

     return response;

}

export const deleteSpecialite=async(specialiteId) =>{

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

        method: 'DELETE'

    });

    const response = await res.json();

    return response;

}

export const addSpecialite=async(specialite)=> {

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

        method: 'POST',

        headers: {

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

        },

        body: JSON.stringify(specialite),

    });

    const response = await res.json();

    return response;

}

export const editSpecialite=async(specialite) =>{

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

        method: 'PUT',

        headers: {

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

        },

        body: JSON.stringify(specialite),

    });

    const response = await res.json();

    return response;

}

# API

Modifier l’API dans route.js sous /src/app/api/specialites



import connectDB from '@/lib/connectDB';

import Specialite from '@/models/Specialite';

import {  NextResponse } from 'next/server';

**import { HttpStatusCode } from 'axios';**

export async function GET() {

    try {

        await connectDB();

        const specialites = await Specialite.find({}, null, {sort: {'\_id': -1}});

        return NextResponse.json(specialites );

    } catch (error) {

        return NextResponse.json({ error });

    }

}

export async function POST(req) {

    try {

        const body = await req.json();

        const newSpecialite = new Specialite(body)

        const specialite = await newSpecialite.save();

        return NextResponse.json(

                { specialite, message: 'Your speciality has been created' },

                { status: HttpStatusCode.Created },

            );

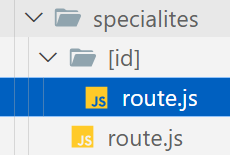
    } catch (error) {

        return NextResponse.json({ message: error }, { status: HttpStatusCode.BadRequest });

    }

}

Puis créer le dossier [id] sous /src/app/api/specialites ayant route.js



import { HttpStatusCode } from 'axios';

import Specialite from '@/models/Specialite';

import {  NextResponse } from 'next/server';

export async function GET(\_, { params }) {

    try {

        const specialite = await Specialite.findById(params.id);

        if (specialite) {

            return NextResponse.json(specialite);

        }

        return NextResponse.json({ message: `Speciality ${params.id} not found` }, { status: HttpStatusCode.NotFound });

    } catch (error) {

        return NextResponse.json({ message: error }, { status: HttpStatusCode.BadRequest });

    }

}

export async function PUT(req, { params }) {

    try {

        const body= await req.json();

        const specialite = await Specialite.findByIdAndUpdate(

            params.id,

            { $set: body },

          { new: true }

        );

            return NextResponse.json({ specialite });

         } catch (error) {

        return NextResponse.json({ message: error }, { status: HttpStatusCode.BadRequest });

    }

}

export async function DELETE(\_, { params }) {

    try {

        await Specialite.findByIdAndDelete(params.id);

        return NextResponse.json({ message: `Speciality ${params.id} has been deleted` });

    } catch (error) {

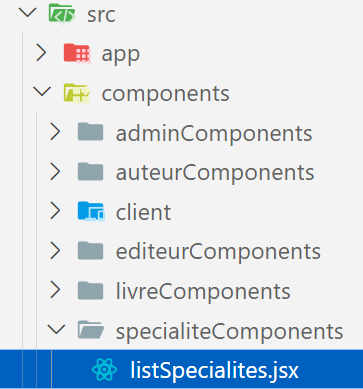
        return NextResponse.json({ message: error }, { status: HttpStatusCode.BadRequest });

    }

}

# Affichage

Créer sous /src/components/ le dossier specialiteComponents ayant listSpecialtes.jsx



'use client'

import React from 'react'

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

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

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

const ListSpecialites = ({specialites}) => {

    const columns = React.useMemo(

        () => [

          {

            accessorKey: 'nomspecialite',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

            Cell: ({ cell, row }) => (

                <div >

                    <button className="btn btn-ghost text-success"

                        onClick={() => {}}

                        size="md"

                    >

                       <EditOutlinedIcon/>

                    </button>

                    <button className="btn btn-ghost text-warning"

                        onClick={() => {}}

                        size="md"

                    >

                          <DeleteForeverIcon />

                    </button>

                </div>

            ),

        },

          ],

        [],

      );

  return (

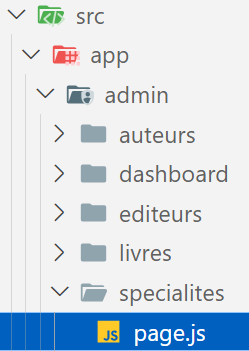
    <MaterialReactTable columns={columns} data={specialites} />

 )

}

export default ListSpecialites

Créer sous /src/app/admin un dossier specialites contenant page.js



import ListSpecialites from "@/components/specialiteComponents/listSpecialites"

import {fetchSpecialites} from "@/services/specialiteService";

async function getData() {

    const data=await fetchSpecialites()

    return data;

  }

  export default async function Page() {

    const data = await getData()

   return (

    <div>

    <ListSpecialites specialites={data} />

    </div>

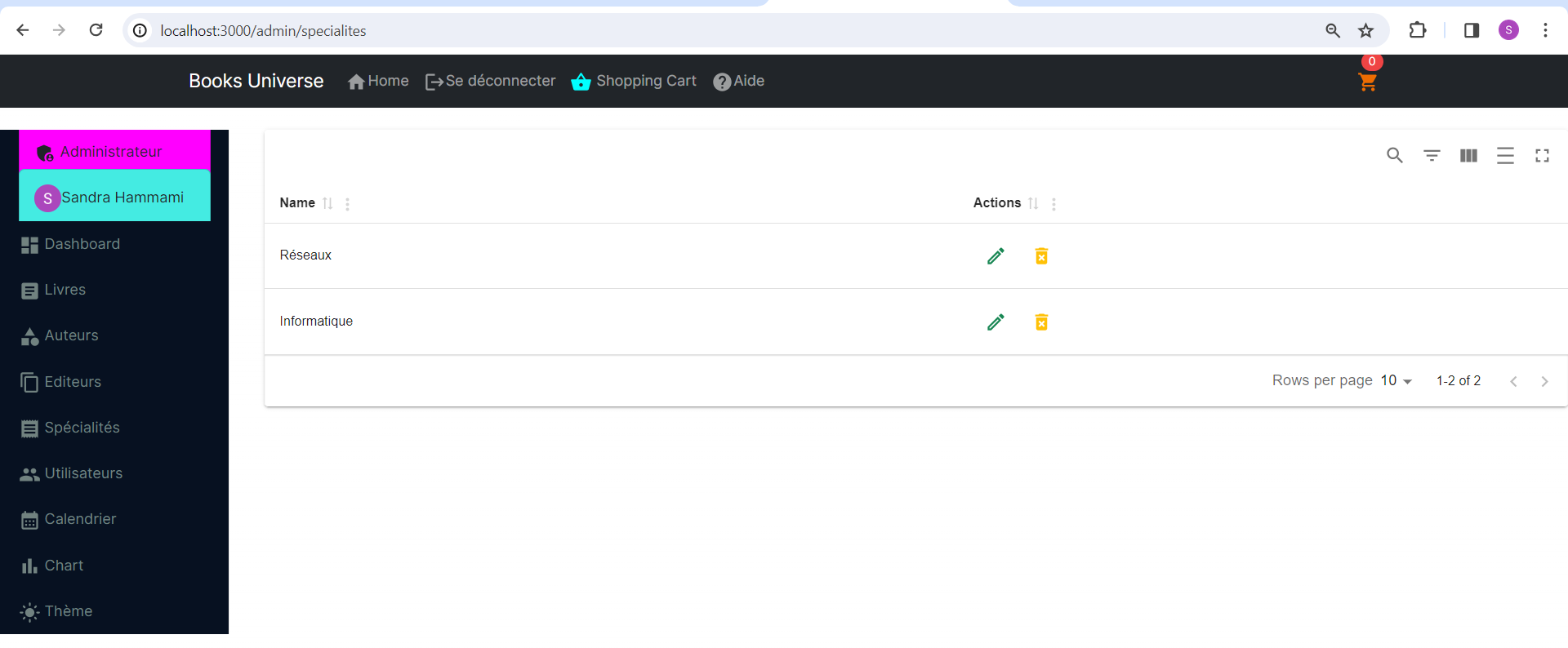
   );

  }

Modifier l’appel de la page dans le SideBar

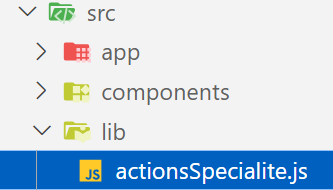


<MenuItem component={<Link href="**/admin/specialites**" />}> <ReceiptIcon/> Spécialités </MenuItem>



# Ajout

Créer sous /src/lib le fichier actionsSpecialite.js



Un Server Action peut être défini avec la directive "use server". Vous pouvez placer la directive en haut d'une fonction asynchrone pour marquer la fonction comme action serveur, ou aussi c’est possible, en haut d'un fichier séparé pour marquer toutes les exportations de ce fichier comme Server Action.

**'use server'**

import { revalidatePath } from 'next/cache'

import {addSpecialite} from '@/services/specialiteService'

export **async** function addSpeciality(prevState, formData) {

    const nomspecialite= formData.get('nomspecialite')

    const specialite ={

        nomspecialite

    }

    try {

        const result = await addSpecialite(specialite);

        revalidatePath('/admin/specialites')

       if(result.specialite.nomspecialite) return { message: `Created speciality ${result.specialite.nomspecialite}` }

       else  return { message: 'Failed to create speciality' }

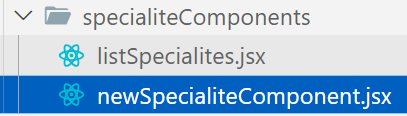
    } catch (e) {

  return { message: 'Error : Failed to create speciality' }

}

}

Créer sous /src/components/specialitesComponents le fichier newSpecialiteComponent.jsx



'use client'

import { useFormState, useFormStatus } from 'react-dom'

import { addSpeciality} from '@/lib/actionsSpecialite'

import { useEffect, useState, useRef } from 'react'

import toast from 'react-hot-toast'

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

import Modal from 'react-bootstrap/Modal';

export default function CreateForm() {

    const [show, setShow] = useState(false);

    const handleClose = () => setShow(false);

    const handleShow = () => setShow(true);

    const [state, formAction] = useFormState(addSpeciality, {

    message: '',

  })

  const { pending } = useFormStatus()

  const ref = useRef()

  useEffect(() => {

   if (state.message){

    setShow(false)

      ref.current?.reset()

      toast(state.message)

    }

  }, [state.message])

  return (

    <div>

      <button

        className="btn btn-dark"

        onClick={handleShow}

        style={{

            textDecoration: 'none',

            color: 'aqua',

            fontSize: 14,

          }}

      >

        <AddCircleOutlineIcon/> Nouveau

      </button>

      <Modal show={show} onHide={handleClose}>

      <form ref={ref} action={formAction}>

        <Modal.Header closeButton>

          <Modal.Title>  <h2 className="tex-2xl font-bold pm-4">Ajout Spécialité</h2> </Modal.Title>

        </Modal.Header>

        <Modal.Body>

            <div className="form-control w-full max-w-xs py-4">

              <label htmlFor="nomspecialite">Nom</label>

              <input

                type="text"

                id="nomspecialite"

                name="nomspecialite"

                className="input input-bordered w-full max-w-xs"

                required

              />

            </div>

            </Modal.Body>

            <Modal.Footer>

            <button

              className="btn btn-primary mr-3"

              type="submit"

              disabled={pending}

            >

              Enregistrer

            </button>

            <button

              type="button"

              className="btn btn-ghost"

              onClick={handleClose}

            >

              Back

            </button>

            </Modal.Footer>

            </form>

          </Modal>

    </div>

  )

}

Modifier /src/app/admin/specialites/page.js



import ListSpecialites from "@/components/specialiteComponents/listSpecialites"

import {fetchSpecialites} from "@/services/specialiteService";

import { Toaster } from 'react-hot-toast'

import CreateForm from '@/components/specialiteComponents/newSpecialiteComponent'

async function getData() {

    const data=await fetchSpecialites()

    return data;

  }

  export default async function Page() {

    const data = await getData()

   return (

    <div>

    <Toaster />

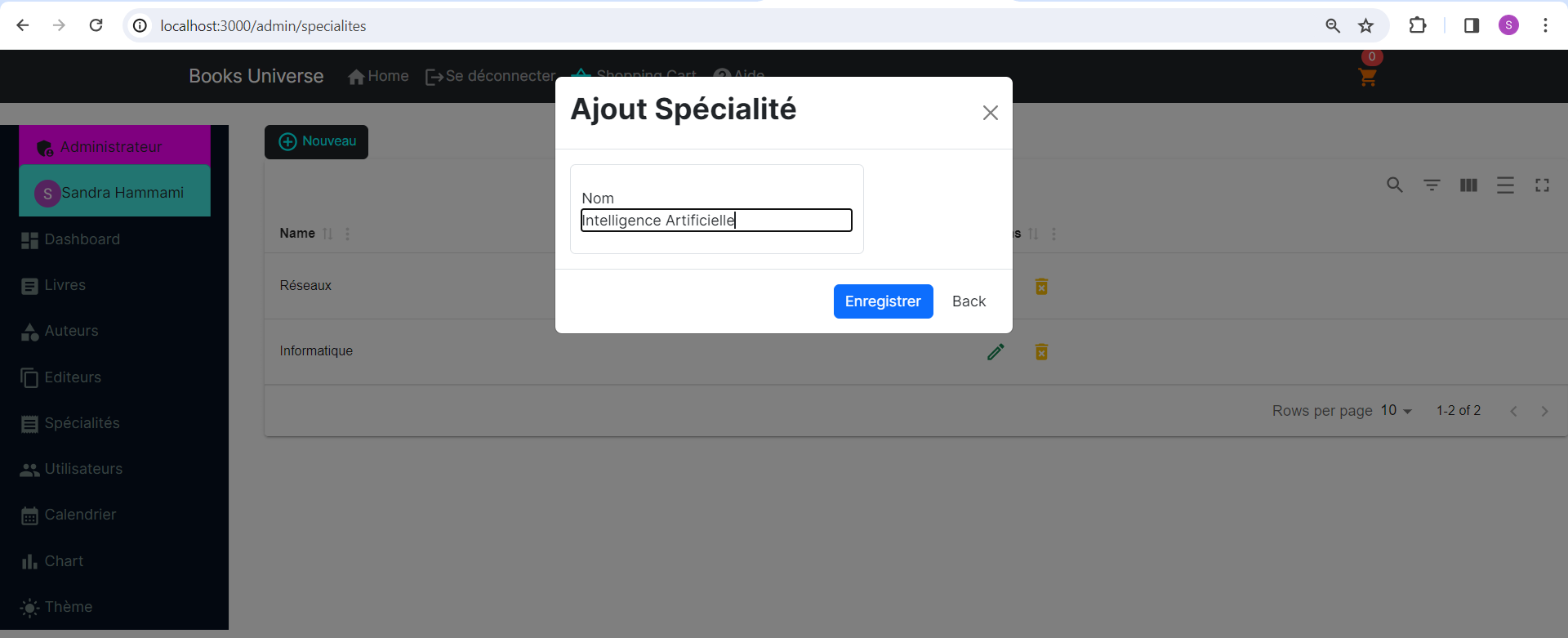
    <CreateForm />

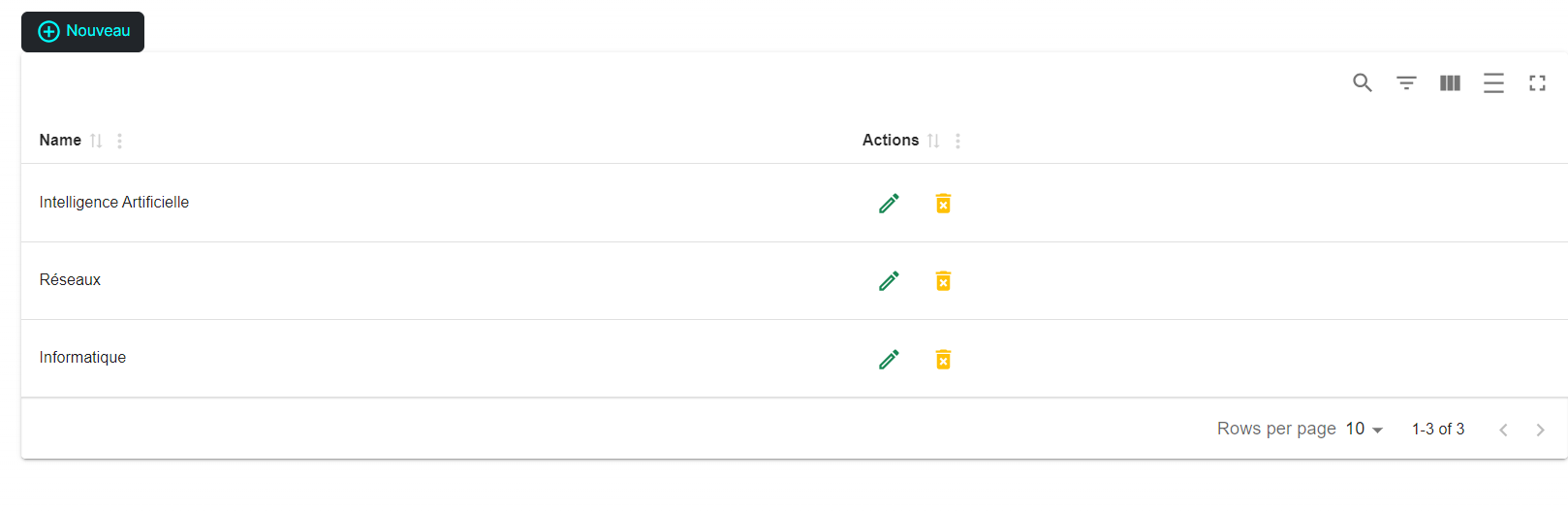
    <ListSpecialites specialites={data} />

    </div>

   );

  }





Suppression

Modifier actionsSpecialite.js



'use server'

import { revalidatePath } from 'next/cache'

import {addSpecialite,**deleteSpecialite**} from '@/services/specialiteService'

export async function addSpeciality(prevState, formData) {

    const nomspecialite= formData.get('nomspecialite')

    const specialite ={

        nomspecialite

    }

    try {

        const result = await addSpecialite(specialite);

        revalidatePath('/admin/specialites')

       if(result.specialite.nomspecialite) return { message: `Created speciality ${result.specialite.nomspecialite}` }

       else  return { message: 'Failed to create speciality' }

    } catch (e) {

  return { message: 'Error : Failed to create speciality' }

}

}

export async function delSpecialite(\_id,nomspecialite){

    try {

        const result = await deleteSpecialite(\_id)

        console.log(result)

         revalidatePath('/admin/specialites')

      return { message: `Deleted speciality ${nomspecialite}` }

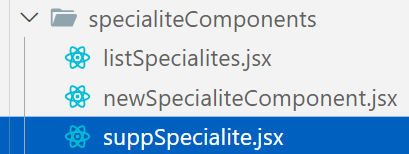
    } catch (e) {

      return { message: 'Failed to delete speciality' }

    }

  }

Créer sous /src/components/specialitesComponents le fichier suppSpecialite.jsx



'use client'

import { delSpecialite} from '@/lib/actionsSpecialite'

import { useFormStatus } from 'react-dom'

import toast from 'react-hot-toast'

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

export default function DeleteForm({ \_id,nomspecialite}) {

  const { pending } = useFormStatus()

  return (

    <form

      action={async () => {

        const res = await delSpecialite(\_id,nomspecialite)

        toast(res.message)

      }}

    >

      <button type="submit" disabled={pending} className="btn btn-ghost text-warning">

      <DeleteForeverIcon />

      </button>

    </form>

  )

}

Puis modifier le contenu de listSpecialites



'use client'

import React from 'react'

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

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

**import DeleteForm from './suppSpecialite';**

const ListSpecialites = ({specialites}) => {

    const columns = React.useMemo(

        () => [

          {

            accessorKey: 'nomspecialite',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

            Cell: ({ cell, row }) => (

                <div >

                    <button className="btn btn-ghost text-success"

                        onClick={() => {}}

                        size="md"

                    >

                       <EditOutlinedIcon/>

                    </button>

                    <DeleteForm

                    \_id={cell.row.original.\_id}

                    nomspecialite={cell.row.original.nomspecialite}

                  />

                </div>

            ),

        },

          ],

        [],

      );

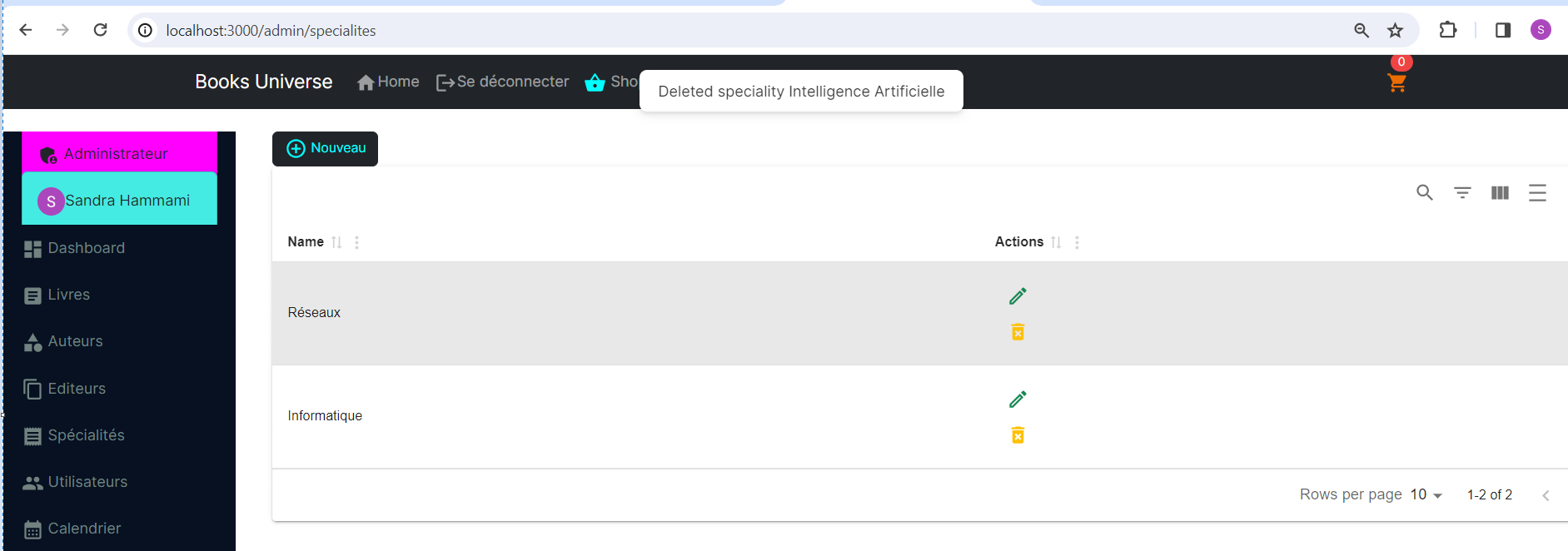
  return (

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

 )

}

export default ListSpecialites



Modification

Modifier actionsSpecialite.js



'use server'

import { revalidatePath } from 'next/cache'

import {addSpecialite,deleteSpecialite,fetchSpecialiteById,editSpecialite} from '@/services/specialiteService'

export async function addSpeciality(prevState, formData) {

    const nomspecialite= formData.get('nomspecialite')

    const specialite ={

        nomspecialite

    }

    try {

        const result = await addSpecialite(specialite);

        revalidatePath('/admin/specialites')

       if(result.specialite.nomspecialite) return { message: `Created speciality ${result.specialite.nomspecialite}` }

       else  return { message: 'Failed to create speciality' }

    } catch (e) {

  return { message: 'Error : Failed to create speciality' }

}

}

export async function delSpecialite(\_id,nomspecialite){

    try {

        const result = await deleteSpecialite(\_id)

        console.log(result)

         revalidatePath('/admin/specialites')

      return { message: `Deleted speciality ${nomspecialite}` }

    } catch (e) {

      return { message: 'Failed to delete speciality' }

    }

  }

  export async function fetchOneSpecialite(\_id) {

    try {

        const result = await fetchSpecialiteById(\_id)

        return result

    } catch (e) {

      return  null

    }

  }

  export async function updateSpecialite(prevState, formData) {

    const \_id= formData.get('\_id')

    const nomspecialite= formData.get('nomspecialite')

    const specialite ={

        \_id,

        nomspecialite

    }

    try {

        const result = await editSpecialite(specialite)

        revalidatePath('/admin/specialites')

       if(result.specialite.nomspecialite) return { message: `Updated Speciality ${result.specialite.nomspecialite}` }

       else  return { message: 'Failed to update Speciality' }

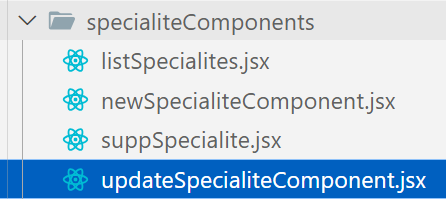
    } catch (e) {

  return { message: 'Failed to update Speciality' }

}

}

Par la suite créer updateSpecialiteComponent.jsx sous /src/components/specialiteComponents



'use client'

import { useFormState, useFormStatus } from 'react-dom'

import { updateSpecialite} from '@/lib/actionsSpecialite'

import { useEffect, useState, useRef } from 'react'

import toast from 'react-hot-toast'

import Modal from 'react-bootstrap/Modal';

export default function EditCategForm({specialite}) {

    const [show, setShow] = useState(false);

    const handleClose = () => setShow(false);

    const [state, formAction] = useFormState(updateSpecialite, {

    message: '',

  })

  const { pending } = useFormStatus()

  const ref = useRef()

  useEffect(() => {

     setShow(true)

 }, [specialite])

  useEffect(() => {

     if (state.message){

      setShow(false)

      ref.current?.reset()

      toast(state.message)

    }

  }, [state.message])

  return (

    <div>

        <Modal show={show} onHide={handleClose}>

      <form ref={ref} action={formAction}>

        <Modal.Header closeButton>

          <Modal.Title>  <h2 className="tex-2xl font-bold pm-4">Modification Spécialité</h2> </Modal.Title>

        </Modal.Header>

        <Modal.Body>

          <input

                type="hidden"

                id="\_id"

                name="\_id"

                value={specialite.\_id}

                />

            <div className="form-control w-full max-w-xs py-4">

              <label htmlFor="nomspecialite">Name</label>

              <input

                type="text"

                id="nomspecialite"

                name="nomspecialite"

                className="input input-bordered w-full max-w-xs"

                required

                defaultValue={specialite.nomspecialite}

              />

            </div>

            </Modal.Body>

            <Modal.Footer>

            <button

              className="btn btn-primary mr-3"

              type="submit"

              disabled={pending}

            >

              Valider

            </button>

            <button

              type="button"

              className="btn btn-ghost"

              onClick={handleClose}

            >

              Back

            </button>

            </Modal.Footer>

            </form>

          </Modal>

    </div>

  )

}

Ensuite Modifier listSpecialites.jsx



'use client'

import React from 'react'

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

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

import DeleteForm from './suppSpecialite';

import UpdateSpecialiteComponent from './updateSpecialiteComponent'

import { fetchOneSpecialite } from '@/lib/actionsSpecialite'

import { useFormStatus } from 'react-dom'

const ListSpecialites = ({specialites}) => {

    const [isEdit,setIsEdit]=React.useState(false)

    const [specialite, setSpecialite]=React.useState("")

    const { pending } = useFormStatus()

    const columns = React.useMemo(

        () => [

          {

            accessorKey: 'nomspecialite',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

            Cell: ({ cell, row }) => (

                <div >

                <form

                action={async () => {

                    const res = await fetchOneSpecialite(cell.row.original.\_id)

                    setSpecialite(res)

                    setIsEdit(true)

                }}

                >

                <button type="submit" disabled={pending} className="btn btn-ghost text-success">

                    <EditOutlinedIcon/>

                </button>

                </form>

                   <DeleteForm

                    \_id={cell.row.original.\_id}

                    nomspecialite={cell.row.original.nomspecialite}

                  />

                </div>

            ),

        },

          ],

        [],

      );

  return (

    <>

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

    {isEdit ? <UpdateSpecialiteComponent specialite={specialite} />

    :null}

    </>

 )

}

export default ListSpecialites

