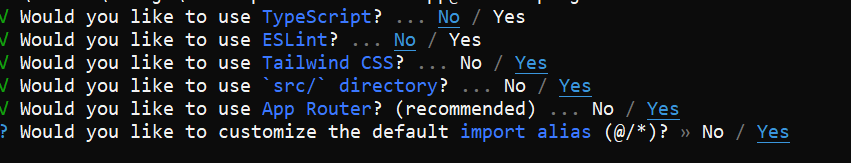
CRUD avec Server actions

Créer projet

npx create-next-app@latest nextjs-project1



Dépendances

npm i material-react-table @mui/material @mui/x-date-pickers @mui/icons-material @emotion/react @emotion/styled

npm i react-hot-toast

Affichage



import ListCategories from "@/components/categories/listCategories"

async function getData() {

    const res = await fetch('http://localhost:3001/api/categories')

    // The return value is \*not\* serialized

    // You can return Date, Map, Set, etc.

    if (!res.ok) {

      // This will activate the closest `error.js` Error Boundary

      throw new Error('Failed to fetch data')

    }

    return res.json()

  }

  export default async function Page() {

    const data = await getData()

   return (

    <main>

    <h1 className="font-bold py-10 text-2xl">Admin Categories</h1>

    <ListCategories categories={data} />

    </main>

   );

  }



'use client'

import React from 'react'

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

const ListCategories = ({categories}) => {

    const columns = React.useMemo(

        () => [

            {

                accessorKey: 'imagecategorie',

                header: 'Image',

                Cell: ({ cell }) => (

                       <img

                            src={cell.getValue()}

                            width={100}

                            height={120}

                            className="rounded-lg"

                            alt="Image Category"

                        />

                    )

                 },

          {

            accessorKey: 'nomcategorie',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

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

                <div >

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

                        onClick={() => {}}

                        size="md"

                    >

                       Edit

                    </button>

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

                        onClick={() => {}}

                        size="md"

                    >

                        Delete

                    </button>

                </div>

            ),

        },

          ],

        [],

      );

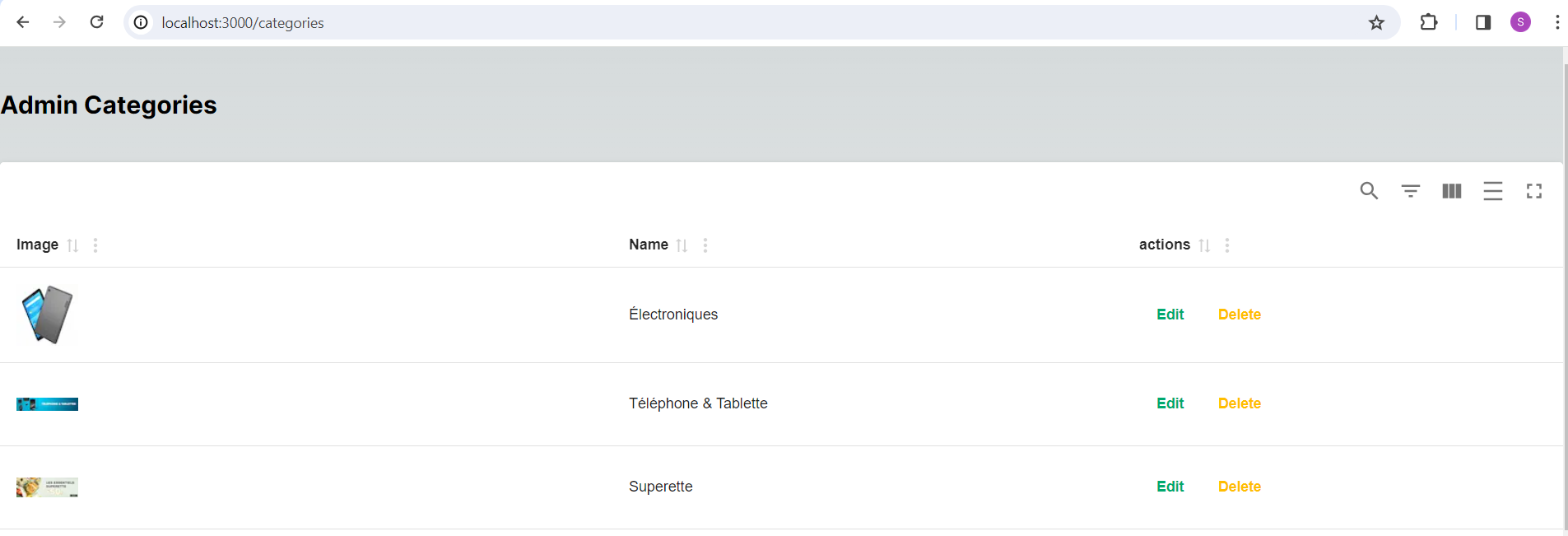
  return (

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

 )

}

export default ListCategories



Ajout



'use client'

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

import { addCategory} from '@/lib/actionsCategorie'

import { useEffect, useRef } from 'react'

import toast from 'react-hot-toast'

export default function CreateForm() {

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

    message: '',

  })

  const { pending } = useFormStatus()

  const ref = useRef()

  useEffect(() => {

   if (state.message){

      (document.getElementById('my\_modal\_3')).close()

      ref.current?.reset()

      toast(state.message)

    }

  }, [state.message])

  return (

    <div>

      <button

        className="btn btn-primary"

        onClick={() =>

          (document.getElementById('my\_modal\_3')).showModal()

        }

      >

        Create Category

      </button>

      <dialog id="my\_modal\_3" className="modal">

        <div className="modal-box">

          <h2 className="tex-2xl font-bold pm-4">Create Category</h2>

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

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

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

              <input

                type="text"

                id="nomcategorie"

                name="nomcategorie"

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

                required

              />

            </div>

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

              <label htmlFor="image">Image</label>

              <input

                type="text"

                id="imagecategorie"

                name="imagecategorie"

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

                required

                defaultValue="https://res.cloudinary.com/iset-sfax/image/upload/v1658754288/images/flash1.jpg.jpg"

              />

            </div>

            <button

              className="btn btn-primary mr-3"

              type="submit"

              disabled={pending}

            >

              Create

            </button>

            <button

              type="button"

              className="btn btn-ghost"

              onClick={() =>

                (document.getElementById('my\_modal\_3')).close()

              }

            >

              Back

            </button>

          </form>

        </div>

      </dialog>

    </div>

  )

}



'use server'

import { revalidatePath } from 'next/cache'

export async function addCategory(prevState, formData) {

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch("http://localhost:3001/api/categories", {

            method: 'POST',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        console.log(result)

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Created category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to create category' }

}

}



import ListCategories from "@/components/categories/listCategories"

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

import CreateForm from '@/components/categories/createCateg'

async function getData() {

    const res = await fetch('http://localhost:3001/api/categories')

    // The return value is \*not\* serialized

    // You can return Date, Map, Set, etc.

    if (!res.ok) {

      // This will activate the closest `error.js` Error Boundary

      throw new Error('Failed to fetch data')

    }

    return res.json()

  }

  export default async function Page() {

    const data = await getData()

   return (

    <main>

    <h1 className="font-bold py-10 text-2xl">Admin Categories</h1>

    <Toaster />

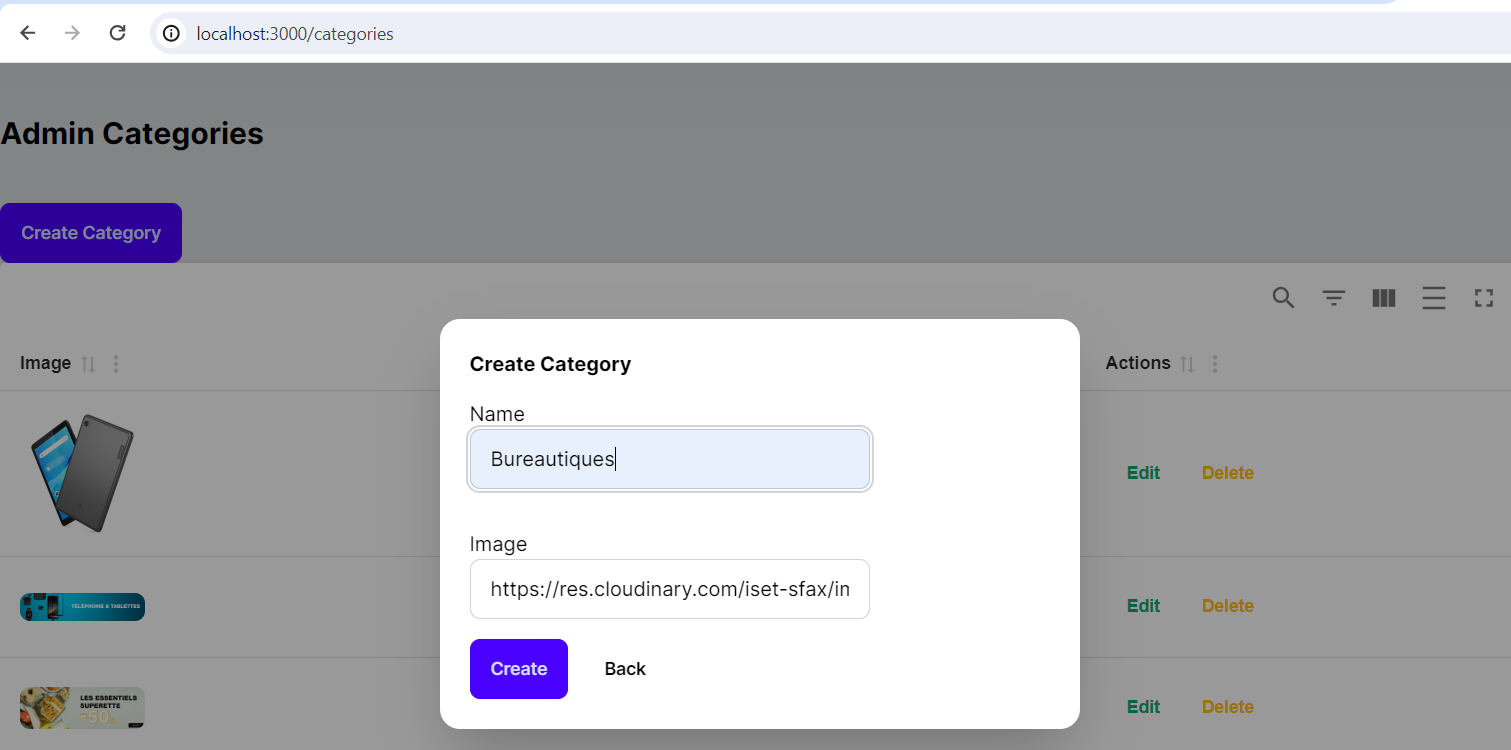
    <CreateForm />

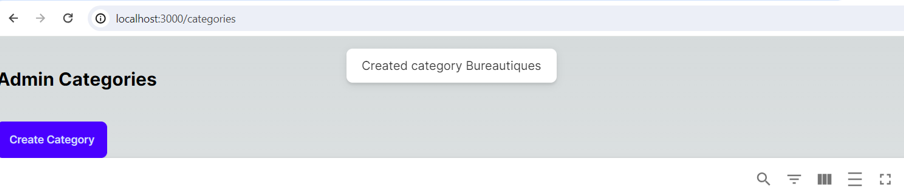
    <ListCategories categories={data} />

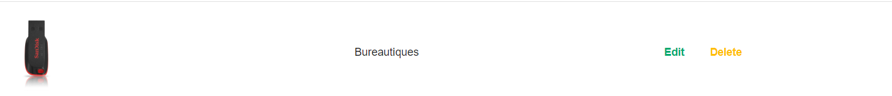
    </main>

   );

  }







Suppression



'use server'

import { revalidatePath } from 'next/cache'

export async function addCategory(prevState, formData) {

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch("http://localhost:3001/api/categories", {

            method: 'POST',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Created category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to create category' }

}

}

export async function deleteCateg(\_id,nomcategorie){

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`,{

            method: 'DELETE'

        });

        const result = await response.json();

        console.log(result)

        revalidatePath('/categories')

      return { message: `Deleted category ${nomcategorie}` }

    } catch (e) {

      return { message: 'Failed to delete category' }

    }

  }



'use client'

import { deleteCateg} from '@/lib/actionsCategorie'

import { useFormStatus } from 'react-dom'

import toast from 'react-hot-toast'

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

  const { pending } = useFormStatus()

  return (

    <form

      action={async () => {

        const res = await deleteCateg(\_id,nomcategorie)

        toast(res.message)

      }}

    >

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

        Delete

      </button>

    </form>

  )

}



'use client'

import React from 'react'

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

import DeleteForm from './deleteCateg'

const ListCategories = ({categories}) => {

    const columns = React.useMemo(

        () => [

            {

                accessorKey: 'imagecategorie',

                header: 'Image',

                Cell: ({ cell }) => (

                       <img

                            src={cell.getValue()}

                            width={100}

                            height={120}

                            className="rounded-lg"

                            alt="Image Category"

                        />

                    )

                 },

          {

            accessorKey: 'nomcategorie',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

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

                <div>

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

                        onClick={() => {}}

                        size="md"

                    >

                       Edit

                    </button>

                    <DeleteForm

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

                    nomcategorie={cell.row.original.nomcategorie}

                  />

                </div>

            ),

        },

          ],

        [],

      );

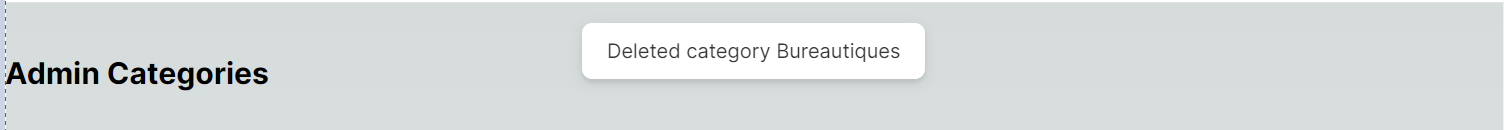
  return (

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

 )

}

export default ListCategories



Modification : Cas d’une page update différente



'use server'

import { revalidatePath } from 'next/cache'

export async function addCategory(prevState, formData) {

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch("http://localhost:3001/api/categories", {

            method: 'POST',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Created category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to create category' }

}

}

export async function deleteCateg(\_id,nomcategorie) {

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`,{

            method: 'DELETE'

        });

        const result = await response.json();

        console.log(result)

        revalidatePath('/categories')

      return { message: `Deleted category ${nomcategorie}` }

    } catch (e) {

      return { message: 'Failed to delete category' }

    }

  }

  export async function updateCategory(prevState, formData) {

    const \_id= formData.get('\_id')

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        \_id,

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`, {

            method: 'PUT',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Updated category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to update category' }

}

}



'use client'

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

import { updateCategory} from '@/lib/actionsCategorie'

import { useEffect, useRef } from 'react'

import toast from 'react-hot-toast'

export default function CreateForm({categorie}) { console.log(categorie)

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

    message: '',

  })

  const { pending } = useFormStatus()

  const ref = useRef()

  useEffect(() => {

   if (state.message){

      ref.current?.reset()

      toast(state.message)

    }

  }, [state.message])

  return (

    <div>

          <h2 className="tex-2xl font-bold pm-4">Update Category</h2>

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

          <input

                type="hidden"

                id="\_id"

                name="\_id"

                value={categorie.\_id}

                />

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

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

              <input

                type="text"

                id="nomcategorie"

                name="nomcategorie"

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

                required

                defaultValue={categorie.nomcategorie}

              />

            </div>

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

              <label htmlFor="image">Image</label>

              <input

                type="text"

                id="imagecategorie"

                name="imagecategorie"

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

                required

                defaultValue={categorie.imagecategorie}

              />

            </div>

            <button

              className="btn btn-primary mr-3"

              type="submit"

              disabled={pending}

            >

              Update

            </button>

          </form>

    </div>

  )

}



Dans votre dossier app/categories, créez une nouvelle route dynamique appelée [\_id] (un dossier), puis une nouvelle route appelée edit (un autre dossier) avec un fichier page.js

import EditCateg from '@/components/categories/editCategPageForm'

export async function editCateg(\_id) {

  try {

      const response = await fetch(`http://localhost:3001/api/categories/${\_id}`,{

          method: 'GET'

      });

      const result = await response.json();

      return result

  } catch (e) {

    return  null

  }

}

export default async function Page({params}) {

  const data = await editCateg(params.\_id)

   return (

    <EditCateg categorie={data} />

  );

}



'use client'

import React from 'react'

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

import Link from 'next/link';

import DeleteForm from './deleteCateg'

const ListCategories = ({categories}) => {

    const columns = React.useMemo(

        () => [

            {

                accessorKey: 'imagecategorie',

                header: 'Image',

                Cell: ({ cell }) => (

                       <img

                            src={cell.getValue()}

                            width={100}

                            height={120}

                            className="rounded-lg"

                            alt="Image Category"

                        />

                    )

                 },

          {

            accessorKey: 'nomcategorie',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

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

                <div>

    <Link

     href={`/categories/${cell.row.original.\_id}/edit`}

     className="btn btn-ghost text-success">

    Edit

    </Link>

                    <DeleteForm

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

                    nomcategorie={cell.row.original.nomcategorie}

                  />

                </div>

            ),

        },

          ],

        [],

      );

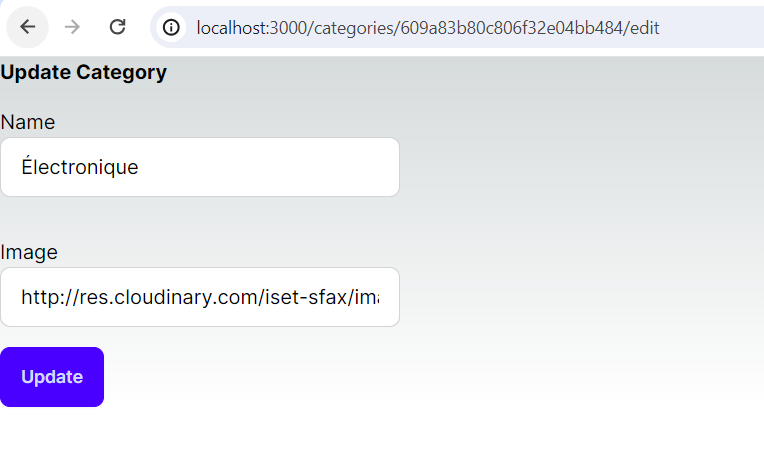
  return (

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

 )

}

export default ListCategories



Modification : Cas d’un Modal



'use server'

import { revalidatePath } from 'next/cache'

export async function addCategory(prevState, formData) {

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch("http://localhost:3001/api/categories", {

            method: 'POST',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Created category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to create category' }

}

}

export async function deleteCateg(\_id,nomcategorie) {

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`,{

            method: 'DELETE'

        });

        const result = await response.json();

        console.log(result)

        revalidatePath('/categories')

      return { message: `Deleted category ${nomcategorie}` }

    } catch (e) {

      return { message: 'Failed to delete category' }

    }

  }

 export async function editCateg(\_id) {

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`,{

            method: 'GET'

        });

        const result = await response.json();

        return result

    } catch (e) {

      return  null

    }

  }

  export async function updateCategory(prevState, formData) {

    const \_id= formData.get('\_id')

    const nomcategorie= formData.get('nomcategorie')

    const imagecategorie= formData.get('imagecategorie')

    const categorie ={

        \_id,

        nomcategorie,

        imagecategorie

    }

    try {

        const response = await fetch(`http://localhost:3001/api/categories/${\_id}`, {

            method: 'PUT',

            headers: {

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

            },

            body: JSON.stringify(categorie),

        });

        const result = await response.json();

        revalidatePath('/categories')

       if(result.nomcategorie) return { message: `Updated category ${result.nomcategorie}` }

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

    } catch (e) {

  return { message: 'Failed to update category' }

}

}



'use client'

import React from 'react'

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

import DeleteForm from './deleteCateg'

import EditCategForm from './editCategForm'

import { editCateg} from '@/lib/actionsCategorie'

import { useFormStatus } from 'react-dom'

const ListCategories = ({categories}) => {

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

  const [categorie, setCategorie]=React.useState("")

  const { pending } = useFormStatus()

    const columns = React.useMemo(

        () => [

            {

                accessorKey: 'imagecategorie',

                header: 'Image',

                Cell: ({ cell }) => (

                       <img

                            src={cell.getValue()}

                            width={100}

                            height={120}

                            className="rounded-lg"

                            alt="Image Category"

                        />

                    )

                 },

          {

            accessorKey: 'nomcategorie',

            header: 'Name',

            size: 150,

          },

          {

            accessorKey: '\_id',

            header: 'Actions',

            size: 100,

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

                <div>

<form

      action={async () => {

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

        setCategorie(res)

        setIsEdit(true)

      }}

    >

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

        Edit

      </button>

    </form>

                    <DeleteForm

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

                    nomcategorie={cell.row.original.nomcategorie}

                  />

                </div>

            ),

        },

          ],

        [],

      );

  return (

    <>

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

     {isEdit ? <EditCategForm categorie={categorie} />

            :null}

    </>

 )

}

export default ListCategories



'use client'

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

import { updateCategory} from '@/lib/actionsCategorie'

import { useEffect, useRef } from 'react'

import toast from 'react-hot-toast'

export default function editCategForm({categorie}) {

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

    message: '',

  })

  const { pending } = useFormStatus()

  const ref = useRef()

  useEffect(() => {

     (document.getElementById('my\_modal\_4')).showModal()

 }, [categorie])

  useEffect(() => {

     if (state.message){

      (document.getElementById('my\_modal\_4')).close()

      ref.current?.reset()

      toast(state.message)

    }

  }, [state.message])

  return (

    <div>

      <dialog id="my\_modal\_4" className="modal">

        <div className="modal-box">

          <h2 className="tex-2xl font-bold pm-4">Update Category </h2>

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

          <input

                type="hidden"

                id="\_id"

                name="\_id"

                value={categorie.\_id}

                />

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

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

              <input

                type="text"

                id="nomcategorie"

                name="nomcategorie"

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

                required

                defaultValue={categorie.nomcategorie}

              />

            </div>

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

              <label htmlFor="image">Image</label>

              <input

                type="text"

                id="imagecategorie"

                name="imagecategorie"

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

                required

                defaultValue={categorie.imagecategorie}

              />

            </div>

            <button

              className="btn btn-primary mr-3"

              type="submit"

              disabled={pending}

            >

              Update

            </button>

            <button

              type="button"

              className="btn btn-ghost"

              onClick={() =>

                {

                  (document.getElementById('my\_modal\_4')).close();

                  ref.current?.reset()

                }

              }

            >

              Back

            </button>

          </form>

        </div>

      </dialog>

    </div>

  )

}

