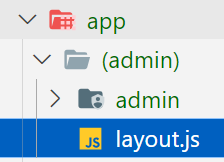
Upload Firebase progress Filepond avec nextJS

# Layout



export const metadata = {

  title: 'Next.js',

  description: 'Generated by Next.js',

}

export default function RootLayout({ children }) {

 return (

    <html lang="en">

   <head>

   <link

        rel="stylesheet"

        href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"

        precedence="default"

      />

    </head>

      <body>Admin pannel

        {children}</body>

    </html>

  )

}

# Axios

Installer

npm i axios

Créer



import axios from 'axios'

export default axios.create({

baseURL:"https://backend-ecommerce-jwt-2024.vercel.app/api"

})

# Services

Créer



import Api from "../Axios/Api";

const ARTICLE\_API="/articles"

export const fetchArticles=async()=> {

return await Api.get(ARTICLE\_API);

}

export const fetchArticleById=async(articleId)=> {

return await Api.get(ARTICLE\_API + '/' + articleId);

}

export const deleteArticle=async(articleId) =>{

return await Api.delete(ARTICLE\_API + '/' + articleId);

}

export const addArticle=async(article)=> {

return await Api.post(ARTICLE\_API, article);

}

export const editArticle=(article) =>{

return Api.put(ARTICLE\_API + '/' + article.\_id, article);

}



import Api from "../Axios/Api";

const SCATEGORIE\_API="/scategories"

export const fetchSCategories=async()=> {

return await Api.get(SCATEGORIE\_API);

}

export const fetchSCategorieById=async(scategorieId)=> {

return await Api.get(SCATEGORIE\_API + '/' + scategorieId);

}

export const deleteSCategorie=async(scategorieId) =>{

return await Api.delete(SCATEGORIE\_API + '/' + scategorieId);

}

export const addSCategorie=async(scategorie)=> {

return await Api.post(SCATEGORIE\_API,scategorie);

}

export const editSCategorie=(scategorie) =>{

return Api.put(SCATEGORIE\_API + '/' + scategorie.\_id, scategorie);

}

# Configurer Firebase

Accéder au site :

<https://console.firebase.google.com/>

# Créer un bucket de stockage cloud

Firebase stocke les fichiers dans un bucket de stockage cloud. Suivez les étapes suivantes pour le créer :

Sur la page de présentation du projet, cliquez sur l'onglet Storage dans le panneau de navigation de gauche.

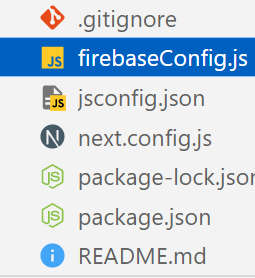


# Ajouter Firebase dans Next JS

Dans votre terminal, accédez au dossier de votre projet. Exécutez la commande suivante pour installer le SDK Firebase :

npm install firebase --legacy-peer-deps

Créez un nouveau fichier appelé firebaseConfig.js dans la racine du projet.



Il faut y coller le code précédemment copié dans les étapes de création du projet dans Firebase. Puis lui ajouter l’apppel de getStorage car on va en avoir besoin pour le stockage des images.

Ce qui donne ce code :

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getStorage } from "firebase/storage";

// Initialize Firebase

const app = initializeApp ({

  apiKey: "AIzaSyATRVSxM4NOndpyJct3th74FY4HQqb-B2I",

  authDomain: "nextproject-76ce9.firebaseapp.com",

  projectId: "nextproject-76ce9",

  storageBucket: "nextproject-76ce9.appspot.com",

  messagingSenderId: "474273333859",

  appId: "1:474273333859:web:ae5786efe351cb8ca66e56"

});

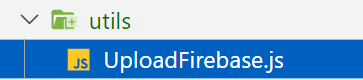
// Firebase storage reference

const storage = getStorage(app);

export default storage;

La dernière ligne exporte la référence de stockage Firebase afin que vous puissiez accéder à cette instance à partir du reste de votre application.

Ensuite, sous le dossier racine créer le dossier utils. Puis il faut y créer le fichier UploadFirebase.js



La méthode Promise.resolve() « résout » une valeur donnée en une promesse. Si la valeur est une promesse, cette promesse est renvoyée.

L'opérateur await permet d'attendre la résolution d'une promesse (Promise). Il ne peut être utilisé qu'au sein d'une fonction asynchrone (définie avec l'instruction async function).

import  storage from "../firebaseConfig";

import { ref, uploadBytesResumable, getDownloadURL } from "firebase/storage";

export const UploadFirebase=async(file)=>{

    let imageurl=""

    const storageRef = ref(storage, `/files/${file.name}`);

    const uploadTask = uploadBytesResumable(storageRef, file);

    await new Promise(resolve => {

    uploadTask.on(

        "state\_changed",

        (snapshot) => {

            const percent = Math.round(

                (snapshot.bytesTransferred / snapshot.totalBytes) \* 100

            );

           console.log(percent);

        },

        (err) => console.log(err),

        async () => {

                await getDownloadURL(uploadTask.snapshot.ref).then((url) => {

                imageurl=url;

                console.log(imageurl);

                resolve();

            });

        }

    );

}

)

    return imageurl;

}

# Formulaire d’ajout

D’abord installer

npm i react-bootstrap bootstrap

On va utiliser Filepond qui est une bibliothèque JavaScript qui peut télécharger tout ce que vous lui lancez, optimise les images pour des téléchargements plus rapides et offre une expérience utilisateur exceptionnelle, accessible et fluide.

<https://www.npmjs.com/package/filepond>

Faire les installations suivantes :

npm i filepond --legacy-peer-deps

npm i react-filepond --legacy-peer-deps

npm i filepond-plugin-image-exif-orientation --legacy-peer-deps

npm i filepond-plugin-image-preview --legacy-peer-deps

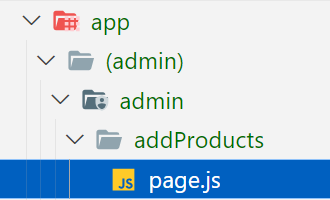
Les images seront sélectionnées puis seront uploadées dans Firebase. Ce dernier nous retournera l’url correspondantes. Pour cela il faut ajouter le site de Firebase dans le tableau domains dans le config pour que l’affichage de ces images soit autorisé.



Ajouter firebasestorage.googleapis.com



Créer



import React from 'react'

const AddProducts = () => {

return (

<div>

page AddProducts

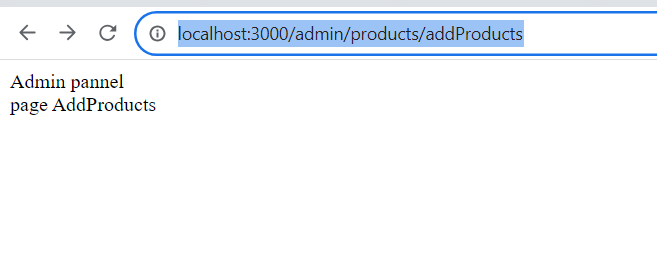
</div>

)

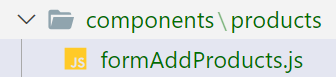
}

export default AddProducts

<http://localhost:3000/admin/products/addProducts>



Créer



"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 { addArticle } from '../../services/ArticleService';

import {UploadFirebase} from '../../utils/UploadFirebase';

import { FilePond,registerPlugin } from 'react-filepond'

import 'filepond/dist/filepond.min.css';

import FilePondPluginImageExifOrientation from 'filepond-plugin-image-exif-orientation'

import FilePondPluginImagePreview from 'filepond-plugin-image-preview'

import 'filepond-plugin-image-preview/dist/filepond-plugin-image-preview.css'

registerPlugin(FilePondPluginImageExifOrientation, FilePondPluginImagePreview)

const Createarticle = ({ scategories }) => {

  const [reference, setReference] = useState("");

  const [designation, setDesignation] = useState("");

  const [prix, setPrix] = useState("");

  const [marque, setMarque] = useState("");

  const [qtestock, setQtestock] = useState("");

  const [imageart, setImageart] = useState("");

  const [scategorieID, setScategorieID] = useState("");

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

  const [file, setFile] = useState("");

  const handleSubmit = (url) => {

    const newProduct = {

      reference,

      designation,

      prix,

      marque,

      qtestock,

      imageart:url,

      scategorieID

    };

//faire le add dans la BD

addArticle(newProduct)

.then(res => {

const response = res.data;

  console.log(response)

    //vider le form

    setReference('');

    setDesignation('');

    setPrix('');

    setMarque('');

    setQtestock('');

    setImageart('');

    setScategorieID('');

    setValidated(false);

    setFile("")

    alert("Added with Success")

  })

  .catch(error=>{

    console.log(error)

    alert("Erreur ! Insertion non effectuée")

    })

}

const handleUpload = (event) => {

  event.preventDefault();

  const form = event.currentTarget;

 if (form.checkValidity() === true) {

        if (!file[0].file) {

            alert("Please upload an image first!");

        }

        else {

          console.log(file[0].file)

          resultHandleUpload(file[0].file,event);

      }

      if (!file[0].file) {

        alert("Please upload an image first!");

    }

    }

 setValidated(true);

};

const resultHandleUpload = async(file) => {

  try {

  const url =  await UploadFirebase(file);

  console.log(url);

  handleSubmit(url)

 } catch (error) {

    console.log(error);

 }

}

const handleReset=()=>{

  setReference('');

    setDesignation('');

    setPrix('');

    setMarque('');

    setQtestock('');

    setImageart('');

    setScategorieID('');

    setValidated(false);

    setFile('')

}

  return (

    <div>

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

  <h2>Create Product</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="6" >

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

  <Form.Control

  required

  type="text"

  placeholder="Référence"

  value={reference}

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

  />

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

  Saisir Référence Article

  </Form.Control.Feedback>

  </Form.Group>

  <Form.Group as={Col} md="6">

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

  <Form.Control

  required

  type="text"

  placeholder="Désignation"

  value={designation}

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

  />

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

  Saisir Désignation

  </Form.Control.Feedback>

  </Form.Group>

  </Row>

  <Row className="mb-2">

  <Form.Group className="col-md-6">

  <Form.Label>Marque \*</Form.Label>

  <InputGroup hasValidation>

  <Form.Control

  type="text"

  required

  placeholder="Marque"

  value={marque}

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

  />

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

  Marque Incorrecte

  </Form.Control.Feedback>

  </InputGroup>

  </Form.Group>

  <Form.Group as={Col} md="6">

<Form.Label>Prix</Form.Label>

<Form.Control

type="number"

placeholder="Prix"

value={prix}

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

/>

</Form.Group>

</Row>

<Row className="mb-3">

<Form.Group className="col-md-6 ">

<Form.Label>

Qté stock<span className="req-tag">\*</span>

</Form.Label>

<Form.Control

required

type="number"

value={qtestock}

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

placeholder="Qté stock"

/>

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

Qté stock Incorrect

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Image</Form.Label>

<FilePond

              files={file}

              allowMultiple={false}

              onupdatefiles={setFile}

              labelIdle='<span class="filepond--label-action">Browse One</span>'

            />

</Form.Group>

<Form.Group as={Col} md="12">

<Form.Label>S/Catégorie</Form.Label>

<Form.Control

as="select"

type="select"

value={scategorieID}

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

>

<option></option>

{scategories.map((scat)=><option key={scat.\_id}

value={scat.\_id}>{scat.nomscategorie}</option>

)}

</Form.Control>

</Form.Group>

</Row>

</div>

</div>

</div>

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

<Button type="button" className="btn btn-warning" onClick={()=>handleReset()}>Annuler</Button>

</Form>

    </div>

  );

};

export default Createarticle;

Modifier :



import React from 'react'

import FormAddProducts from '../../../../../components/products/formAddProducts';

import { fetchSCategories } from '../../../../../services/ScategorieService';

async function getScategories(){

      const scategories = await fetchSCategories();

      return scategories;

  }

const AddProducts = async() => {

    const scategories = await getScategories();

return (

<div>

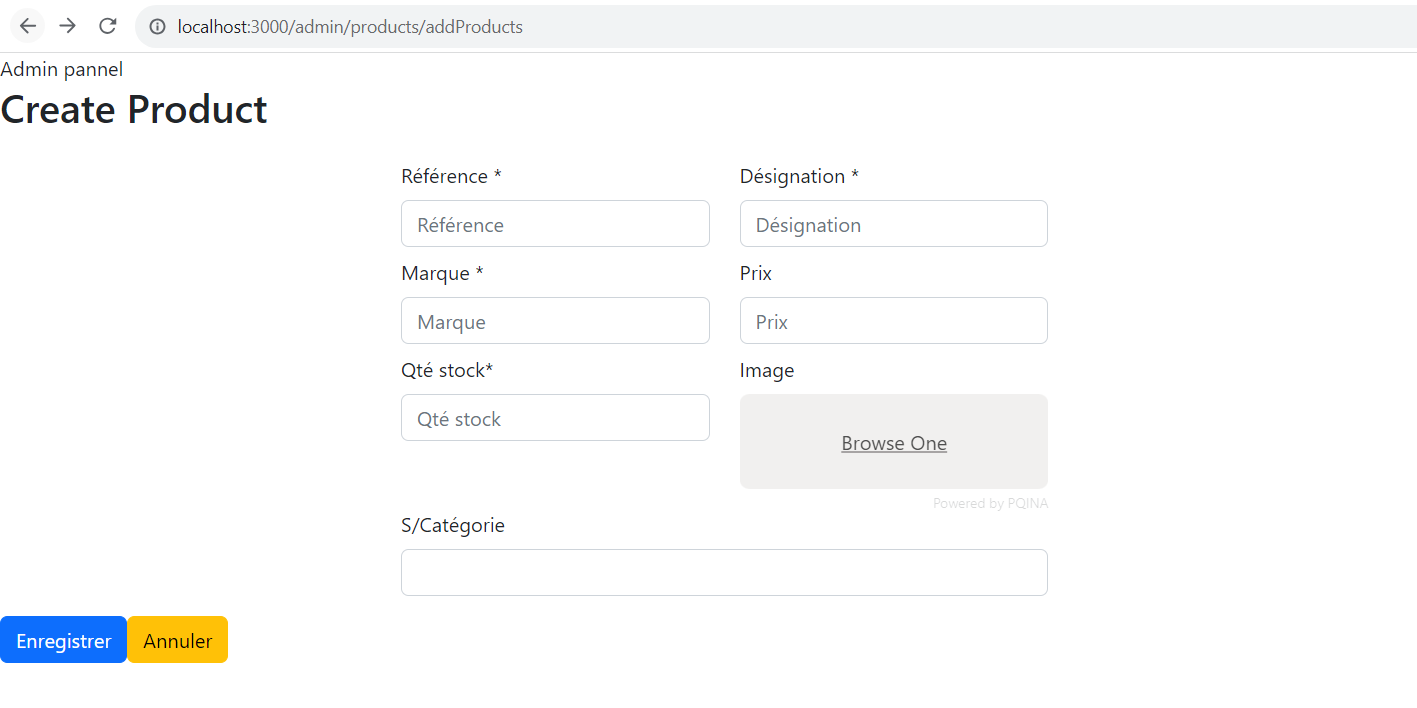
<FormAddProducts scategories = {scategories} />

</div>

)

}

export default AddProducts



# Amélioration : Progress Firebase dans Filepond

Modifier :



"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 { addArticle } from '../../services/ArticleService';

import {UploadFirebase} from '../../utils/UploadFirebase';

import { FilePond,registerPlugin } from 'react-filepond'

import 'filepond/dist/filepond.min.css';

import FilePondPluginImageExifOrientation from 'filepond-plugin-image-exif-orientation'

import FilePondPluginImagePreview from 'filepond-plugin-image-preview'

import 'filepond-plugin-image-preview/dist/filepond-plugin-image-preview.css'

registerPlugin(FilePondPluginImageExifOrientation, FilePondPluginImagePreview)

const Createarticle = ({ scategories }) => {

  const [reference, setReference] = useState("");

  const [designation, setDesignation] = useState("");

  const [prix, setPrix] = useState("");

  const [marque, setMarque] = useState("");

  const [qtestock, setQtestock] = useState("");

  const [imageart, setImageart] = useState("");

  const [scategorieID, setScategorieID] = useState("");

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

  const [file, setFile] = useState("");

  const handleSubmit = (url) => {

    const newProduct = {

      reference,

      designation,

      prix,

      marque,

      qtestock,

**imageart,**

      scategorieID

    };

//faire le add dans la BD

addArticle(newProduct)

.then(res => {

const response = res.data;

  console.log(response)

    //vider le form

    setReference('');

    setDesignation('');

    setPrix('');

    setMarque('');

    setQtestock('');

    setImageart('');

    setScategorieID('');

    setValidated(false);

    setFile("")

    alert("Added with Success")

  })

  .catch(error=>{

    console.log(error)

    alert("Erreur ! Insertion non effectuée")

    })

}

const handleUpload = (event) => {

    event.preventDefault();

    const form = event.currentTarget;

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

          if (!imageart) {

              alert("Please upload an image first!");

          }

          else {

            handleSubmit()

        }

   setValidated(true);

  };

  }

  const resultHandleUpload = async(file,load) => {

    try {

    const url =  await UploadFirebase(file);

    console.log(url)

   if (url){

    load(url);

    setImageart(url)

   }

   } catch (error) {

      console.log(error);

   }

  }

  const process = (

    fieldName,

      file,

      metadata,

      load,

      error,

      progress,

      abort,

      transfer,

      options

    ) => {

      resultHandleUpload(file,load);

      }

const handleReset=()=>{

  setReference('');

    setDesignation('');

    setPrix('');

    setMarque('');

    setQtestock('');

    setImageart('');

    setScategorieID('');

    setValidated(false);

    setFile('')

}

  return (

    <div>

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

  <h2>Create Product</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="6" >

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

  <Form.Control

  required

  type="text"

  placeholder="Référence"

  value={reference}

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

  />

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

  Saisir Référence Article

  </Form.Control.Feedback>

  </Form.Group>

  <Form.Group as={Col} md="6">

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

  <Form.Control

  required

  type="text"

  placeholder="Désignation"

  value={designation}

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

  />

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

  Saisir Désignation

  </Form.Control.Feedback>

  </Form.Group>

  </Row>

  <Row className="mb-2">

  <Form.Group className="col-md-6">

  <Form.Label>Marque \*</Form.Label>

  <InputGroup hasValidation>

  <Form.Control

  type="text"

  required

  placeholder="Marque"

  value={marque}

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

  />

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

  Marque Incorrecte

  </Form.Control.Feedback>

  </InputGroup>

  </Form.Group>

  <Form.Group as={Col} md="6">

<Form.Label>Prix</Form.Label>

<Form.Control

type="number"

placeholder="Prix"

value={prix}

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

/>

</Form.Group>

</Row>

<Row className="mb-3">

<Form.Group className="col-md-6 ">

<Form.Label>

Qté stock<span className="req-tag">\*</span>

</Form.Label>

<Form.Control

required

type="number"

value={qtestock}

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

placeholder="Qté stock"

/>

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

Qté stock Incorrect

</Form.Control.Feedback>

</Form.Group>

<Form.Group as={Col} md="6">

<Form.Label>Image</Form.Label>

<FilePond

              files={file}

              allowMultiple={false}

              onupdatefiles={setFile}

              labelIdle='<span class="filepond--label-action">Browse One</span>'

              server={{ process }}

              name="file"

            />

</Form.Group>

<Form.Group as={Col} md="12">

<Form.Label>S/Catégorie</Form.Label>

<Form.Control

as="select"

type="select"

value={scategorieID}

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

>

<option></option>

{scategories.map((scat)=><option key={scat.\_id}

value={scat.\_id}>{scat.nomscategorie}</option>

)}

</Form.Control>

</Form.Group>

</Row>

</div>

</div>

</div>

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

<Button type="button" className="btn btn-warning" onClick={()=>handleReset()}>Annuler</Button>

</Form>

    </div>

  );

};

export default Createarticle;

