

Ce guide couvre la configuration du frontend Next.js, du backend Django, et l'intégration avec Firebase.

1. Configurer Firebase dans le Projet Next.js

Installer les SDK Firebase

Installe les packages nécessaires via npm :

```
1  npm install firebase
```

Configurer Firebase

Crée un fichier de configuration Firebase (`firebaseConfig.js`) dans ton projet Next.js :

```
1  // frontend/firebaseConfig.js
2  import { initializeApp } from "firebase/app";
3  import { getStorage } from "firebase/storage";
4
5  // Configuration Firebase
6  const firebaseConfig = {
7    apiKey: "AIzaSyApT9doUgp4W_Sk9OCTYLXlapj7I0M1Ez8",
8    authDomain: "uni-map-c7975.firebaseio.com",
9    projectId: "uni-map-c7975",
10   storageBucket: "uni-map-c7975.appspot.com",
11   messagingSenderId: "160778039644",
12   appId: "1:160778039644:web:02f91d9fc5fdb041159199"
13 };
14
15 // Initialiser Firebase
16 const app = initializeApp(firebaseConfig);
17 const storage = getStorage(app);
18
19 export { storage };
```

2. Configurer le Backend Django

Installer `firebase-admin`

Installe le package `firebase-admin` :

```
1 pip install firebase-admin
```

Configurer Firebase Admin SDK

Crée un fichier de configuration Firebase Admin SDK (`firebase_config.py`) (généralement en téléchargeant un fichier de clé de service JSON depuis la console Firebase).

:

```
1 # backend/firebase_config.py
2 import firebase_admin
3 from firebase_admin import credentials, storage
4
5 cred = credentials.Certificate('path/to/serviceAccountKey.json')
6 firebase_admin.initialize_app(cred, {
7     'storageBucket': 'uni-map-c7975.appspot.com'
8 })
9
10 bucket = storage.bucket()
```

Gérer le Téléversement et la Simplification des Fichiers

1. Recevoir et Simplifier le Fichier

Crée une vue Django pour recevoir le fichier, le simplifier, et le téléverser dans Firebase Storage :

```
1 # backend/views.py
2 from rest_framework.decorators import api_view
3 from rest_framework.response import Response
4 from rest_framework import status
5 from django.core.files.storage import default_storage
6 from .firebase_config import bucket
7 import os
8 import datetime
9
10 @api_view(['POST'])
11 def simplify_3d_model(request):
12     file = request.FILES.get('file')
13     user_id = request.data.get('userId')
14
15     if not file:
16         return Response({'error': 'No file provided'},
17                         status=status.HTTP_400_BAD_REQUEST)
18
19     # Simplifier le modèle 3D (implémentez votre logique ici)
```

```

19     simplified_file_path = '/tmp/' + file.name
20     with open(simplified_file_path, 'wb') as f:
21         for chunk in file.chunks():
22             f.write(chunk)
23
24     # Télécharger sur Firebase Storage
25     blob = bucket.blob(f"simplified_models/{user_id}/{file.name}")
26     blob.upload_from_filename(simplified_file_path)
27
28     # Créer une URL de téléchargement
29     download_url =
blob.generate_signed_url(expiration=datetime.timedelta(days=7))
30
31     # Nettoyer le fichier local
32     os.remove(simplified_file_path)
33
34     return Response({'downloadUrl': download_url})

```

2. Ajouter les URLs d'API

Assure-toi que tes URLs d'API sont bien configurées pour accéder à cette vue :

```

1  # backend/urls.py
2  from django.urls import path
3  from . import views
4
5  urlpatterns = [
6      path('api/simplify3d/', views.simplify_3d_model,
7          name='simplify_3d_model'),
8  ]

```

3. Adapter le Frontend Next.js

Modifie le frontend pour gérer l'upload du fichier et afficher l'URL de téléchargement du fichier simplifié :

```

1  "use client";
2  import { useState, useEffect, ChangeEvent } from "react";
3  import ThreeDViewer from './ThreeDViewer';
4  import Footer from "@components/main/Footer";
5  import Navbar2 from "@components/main/Navbar2";
6
7  const Traitement3D: React.FC = () => {
8      const [displayedText, setDisplayedText] = useState<string>('');
9      const fullText: string = "Veuillez importer le fichier que vous voulez
traiter";
10     const typingSpeed: number = 10;

```

```

11
12   useEffect(() => {
13     let index: number = 0;
14     const interval = setInterval(() => {
15       setDisplayedText((prev) => prev + fullText[index]);
16       index++;
17       if (index === fullText.length - 1) {
18         clearInterval(interval);
19       }
20     }, typingSpeed);
21     return () => clearInterval(interval);
22   }, []);
23
24   const [file, setFile] = useState<File | null>(null);
25   const [fileUrl, setFileUrl] = useState<string | null>(null);
26   const [showViewer, setShowViewer] = useState<boolean>(false);
27   const [downloadUrl, setDownloadUrl] = useState<string | null>(null);
28
29   const handleFileChange = (event: ChangeEvent<HTMLInputElement>) => {
30     const file = event.target.files?.[0];
31     if (file) {
32       setFile(file);
33       const url = URL.createObjectURL(file);
34       setFileUrl(url);
35       setShowViewer(false);
36     } else {
37       setFile(null);
38       setFileUrl(null);
39       setShowViewer(false);
40     }
41   };
42
43   const handleVisualiserClick = () => {
44     if (fileUrl) {
45       setShowViewer(true);
46     }
47   };
48
49   const handleSimplifierClick = async () => {
50     if (file) {
51       const formData = new FormData();
52       formData.append('file', file);
53       formData.append('userId', 'exampleUserId'); // Remplace par l'ID
utilisateur réel
54
55       const response = await fetch('/api/simplify3d/', {
56         method: 'POST',
57         body: formData,
58       });
59

```

```

60     if (response.ok) {
61         const data = await response.json();
62         setDownloadUrl(data.downloadUrl);
63     } else {
64         console.error('Failed to simplify the file');
65     }
66 }
67 };
68
69 return (
70     <div className="min-h-screen flex flex-col bg-grenn">
71         <Navbar2 />
72         <main className="flex flex-col items-center justify-center flex-1 p-4">
73             <h2 className="text-4xl font-extrabold mb-6 text-custom-grey shadow-md
p-2 rounded-md">
74                 {displayedText}
75             </h2>
76             <div className="flex items-center space-x-4 my-14 w-150">
77                 <label htmlFor="file-upload" className="bg-custom-green w-40 text-
white text-center px-4 py-2 shadow-md rounded-lg cursor-pointer">
78                     Choose file
79                 </label>
80                 <input
81                     id="file-upload"
82                     type="file"
83                     accept=".stl,.obj" // Ajoute les types de fichiers acceptés ici
84                     className="hidden border-3 border-black rounded-20 p-2 mb-4"
85                     onChange={handleFileChange}
86                 />
87                 <span className="border border-black text-center shadow-md px-4 py-2
rounded-md text-black-900 w-80">
88                     {file ? file.name : 'No file chosen'}
89                 </span>
90             </div>
91
92             <div className="flex space-x-14 mt-10 mb-20">
93                 <button className="bg-custom-green text-white px-6 py-4 rounded
hover:bg-blue-600" onClick={handleVisualiserClick}>Visualiser</button>
94                 <button className="bg-custom-green text-white px-6 py-4 rounded
hover:bg-green-600" onClick={handleSimplifierClick}>Simplifier</button>
95             </div>
96
97             {showViewer && fileUrl && (
98                 <div className="w-full max-w-4xl mt-10 p-4 border border-gray-300
rounded-md shadow-lg bg-beige">
99                     <h3 className="text-xl font-semibold mb-4 text-center">3D Model
Preview</h3>
100                     <div className="w-full h-[500px]">
101                         <ThreeDViewer modelUrl={fileUrl} />
102                     </div>

```

```

103         </div>
104     })
105
106     {downloadUrl && (
107         <div className="mt-10">
108             <a href={downloadUrl} className="bg-custom-green text-white px-6
109             py-4 rounded hover:bg-yellow-600" download>Download Simplified Model</a>
110         </div>
111     })
112     </main>
113     <Footer />
114 </div>
115 );
116 };
117 export default Traitement3D;

```

Résumé

1. Frontend Next.js :

- Installe et configure Firebase pour le stockage des fichiers.
- Gère l'upload du fichier 3D et envoie-le au backend pour simplification.
- Affiche le modèle 3D et fournit un lien pour télécharger le fichier simplifié.

2. Backend Django :

- Configure Firebase Admin SDK pour interagir avec Firebase Storage.
- Crée une vue pour simplifier le modèle 3D, le téléverser sur Firebase, et renvoyer une URL de téléchargement.

3. Configuration :

- Assure-toi que les clés Firebase et les configurations sont correctement définies et sécurisées.