

# Guide Complet

## 1. Configuration du Backend (Django)

### a. Installation des Dépendances

Assure-toi que tu as installé les bibliothèques nécessaires pour utiliser Firebase, simplifier les fichiers 3D et gérer les fichiers avec Django. Tu peux le faire via `pip` :

```
1 pip install firebase-admin trimesh djangorestframework
```

### b. Configuration de Firebase

1. Crée un projet Firebase et configure Firebase Storage et Firestore.
2. Télécharge les informations d'identification du service Firebase et place-les dans un fichier, par exemple `serviceAccountKey.json`.

### c. Configuration de Firebase dans Django

Crée un fichier `firebase.py` pour initialiser Firebase :

```
1 # backend/firebase.py
2 import firebase_admin
3 from firebase_admin import credentials, firestore, storage
4
5 # Initialiser Firebase avec les informations d'identification
6 cred = credentials.Certificate('path/to/serviceAccountKey.json')
7 firebase_admin.initialize_app(cred, {
8     'storageBucket': 'your-project-id.appspot.com'
9 })
10
11 # Référence à Firestore
12 db = firestore.client()
13
14 # Référence à Firebase Storage
15 bucket = storage.bucket()
```

### d. Traitement des Fichiers

Crée un fichier `simplification.py` pour la simplification du fichier 3D :

```

1  # backend/simplification.py
2  import trimesh
3
4  def simplify_mesh(input_file, output_file, simplification_ratio=0.5):
5      mesh = trimesh.load(input_file)
6      simplified_mesh = mesh.simplify_quadratic_decimation(int(len(mesh.faces) *
7      simplification_ratio))
8      simplified_mesh.export(output_file)

```

## e. Gestion des Téléchargements et des Métadonnées

Crée un fichier `file_management.py` pour télécharger le fichier simplifié et stocker les métadonnées dans Firestore :

```

1  # backend/file_management.py
2  import os
3  from firebase import storage, firestore
4
5  def upload_file_to_firebase(file_path, file_name):
6      """Télécharger le fichier simplifié dans Firebase Storage et obtenir l'URL
7      de téléchargement"""
8      blob = bucket.blob(file_name)
9      blob.upload_from_filename(file_path)
10     return blob.public_url
11
12 def store_file_metadata(file_name, download_url, user_id):
13     """Stocker les métadonnées du fichier dans Firestore"""
14     file_metadata = {
15         'fileName': file_name,
16         'downloadUrl': download_url,
17         'userId': user_id,
18         'uploadDate': firestore.SERVER_TIMESTAMP,
19     }
20     db.collection('files').add(file_metadata)

```

## f. Vue API pour le Traitement des Fichiers

Crée une vue API dans `views.py` pour gérer le téléchargement et la simplification des fichiers :

```

1  # backend/views.py
2  from rest_framework.views import APIView
3  from rest_framework.response import Response
4  from rest_framework import status
5  from .file_management import upload_file_to_firebase, store_file_metadata
6  from .simplification import simplify_mesh

```

```

7
8 class Simplify3DView(APIView):
9     def post(self, request):
10         input_file = request.FILES.get('file')
11         user_id = request.data.get('userId') # Assumes the user ID is sent in
the request body
12
13         if input_file:
14             output_file = 'backend/output_file.fbx' # Chemin pour enregistrer
le fichier simplifié
15
16             # Simplification du fichier 3D
17             simplify_mesh(input_file, output_file, simplification_ratio=0.5)
18
19             # Télécharger le fichier simplifié dans Firebase Storage
20             download_url = upload_file_to_firebase(output_file,
'simplified_output_file.fbx')
21
22             # Stocker les métadonnées dans Firestore
23             store_file_metadata('simplified_output_file.fbx', download_url,
user_id)
24
25             # Réponse avec l'URL de téléchargement
26             return Response({'message': 'Simplification réussie',
'downloadUrl': download_url}, status=status.HTTP_200_OK)
27         else:
28             return Response({'error': 'Aucun fichier fourni'},
status=status.HTTP_400_BAD_REQUEST)

```

## 2. Configuration du Frontend (Next.js)

### a. Installation des Dépendances

Assure-toi d'avoir installé Axios pour gérer les requêtes HTTP :

```

1 npm install axios

```

### b. Formulaire de Téléchargement et de Traitement

Modifie ton composant `Traitement3D.tsx` pour gérer le téléchargement et la simplification des fichiers :

```

1
2
3 "use client";
4 import { useState, useEffect, ChangeEvent } from "react";

```

```

5 import ThreeDViewer from './ThreeDViewer';
6 import Footer from "@components/main/Footer";
7 import Navbar2 from "@components/main/Navbar2";
8 import axios from 'axios';
9
10 const Traitement3D: React.FC = () => {
11     const [displayedText, setDisplayedText] = useState<string>('');
12     const fullText: string = "Veuillez importer le fichier que vous voulez
    traiter";
13     const typingSpeed: number = 10; // Vitesse de la frappe en millisecondes
14
15     useEffect(() => {
16         let index: number = 0;
17         const interval = setInterval(() => {
18             setDisplayedText((prev) => prev + fullText[index]);
19             index++;
20             if (index === fullText.length - 1) {
21                 clearInterval(interval);
22             }
23         }, typingSpeed);
24         return () => clearInterval(interval);
25     }, []);
26
27     const [file, setFile] = useState<File | null>(null);
28     const [fileName, setFileName] = useState<string>('');
29     const [fileUrl, setFileUrl] = useState<string | null>(null);
30     const [showViewer, setShowViewer] = useState<boolean>(false);
31     const [simplifiedFileUrl, setSimplifiedFileUrl] = useState<string | null>
    (null);
32
33     const handleFileChange = (event: ChangeEvent<HTMLInputElement>) => {
34         const file = event.target.files?.[0];
35         if (file) {
36             setFile(file);
37             setFileName(file.name);
38             setFileUrl(URL.createObjectURL(file));
39             setShowViewer(false); // Reset the viewer when a new file is selected
40         } else {
41             setFile(null);
42             setFileName('');
43             setFileUrl(null);
44             setShowViewer(false); // Reset the viewer when no file is selected
45         }
46     };
47
48     const handleVisualiserClick = () => {
49         if (fileUrl) {
50             setShowViewer(true);
51         }
52     };

```

```

53
54     const handleSimplifierClick = async () => {
55         if (file) {
56             const formData = new FormData();
57             formData.append('file', file);
58             formData.append('userId', 'some-user-id'); // Remplace par l'ID de
l'utilisateur actuel
59
60             try {
61                 const response = await axios.post('/api/simplify3d', formData, {
62                     headers: {
63                         'Content-Type': 'multipart/form-data',
64                     },
65                 });
66
67                 if (response.data.downloadUrl) {
68                     setSimplifiedFileUrl(response.data.downloadUrl);
69                     setShowViewer(false); // Hide the viewer if you want
70                 }
71             } catch (error) {
72                 console.error('Erreur lors du téléchargement du fichier : ', error);
73             }
74         }
75     };
76
77     return (
78         <div className="min-h-screen flex flex-col bg-grenn">
79             <Navbar2/>
80             <main className="flex flex-col items-center justify-center flex-1 p-4">
81                 <h2 className="text-4xl font-extrabold mb-6 text-custom-grey shadow-md
p-2 rounded-md">
82                     {displayedText}
83                 </h2>
84                 <div className="flex items-center space-x-4 my-14 w-150">
85                     <label htmlFor="file-upload" className="bg-custom-green w-40 text-
white text-center px-4 py-2 shadow-md rounded-lg cursor-pointer">
86                         Choose file
87                     </label>
88                     <input
89                         id="file-upload"
90                         type="file" accept="model/*" // Assure que le fichier est bien un
modèle 3D
91                         className="hidden border-3 border-black rounded-20 p-2 mb-4"
92                         onChange={handleFileChange}
93                     />
94                     <span className="border border-black text-center shadow-md px-4 py-2
rounded-md text-black-900 w-80">
95                         {fileName || 'No file chosen'}
96                     </span>
97                 </div>

```

```

98
99     <div className="flex space-x-14 mt-10 mb-20">
100         <button className="bg-custom-green text-white px-6 py-4 rounded
101         hover:bg-blue-600" onClick={handleVisualiserClick}>Visualiser</button>
102         <button className="bg-custom-green text-white px-6 py-4 rounded
103         hover:bg-green-600" onClick={handleSimplifierClick}>Simplifier</button>
104         { /* Autres boutons */ }
105     </div>
106
107     {showViewer && fileUrl && (
108         <div className="w-full max-w-4xl mt-10 p-4 border border-gray-300
109         rounded-md shadow-lg bg-beige">
110             <h3 className="text-xl font-semibold mb-4 text-center">3D Model
111             Preview</h3>
112             <div className="w-full h-[500px]">
113                 <ThreeDViewer modelUrl={fileUrl} />
114             </div>
115         </div>
116     )}
117
118     {simplifiedFileUrl && (
119         <div className="mt-10">
120             <a href={simplifiedFileUrl} download className="bg-custom-green
121             text-white px-6 py-4 rounded hover:bg-yellow-600">
122                 Télécharger le fichier simplifié
123             </a>
124         </div>
125     )}
126
127 </main>
128 <Footer/>
129 </div>
130
131 );
132 };
133
134 export default Traitement3D;

```

### c. API Route dans Next.js

Crée une API route dans Next.js pour rediriger les requêtes vers ton backend Django :

```

1  // frontend/pages/api/simplify3d.js
2
3  import axios from 'axios';
4
5  export default async function handler(req, res) {
6      if (req.method === 'POST') {
7          try {

```

```

8      const response = await
    axios.post('http://localhost:8000/api/simplify3d/', req.body, {
9          headers: {
10             'Content-Type': 'multipart/form-data',
11          },
12      });
13
14      res.status(response.status).json(response.data);
15  } catch (error) {
16      res.status(error.response?.status || 500).json({ error: error.message
17  });
18  }
19  } else {
20      res.setHeader('Allow', ['POST']);
21      res.status(405).end(`Method ${req.method} Not Allowed`);
22  }
23

```

## Récapitulatif

### 1. Frontend (Next.js) :

- Permet aux utilisateurs de sélectionner un fichier 3D.
- Envoie le fichier au backend via une API route Next.js.

### 2. Backend (Django) :

- Reçoit le fichier, le simplifie et le stocke dans Firebase Storage.
- Stocke les métadonnées dans Firestore.
- Renvoie l'URL du fichier simplifié au frontend.

### 3. Stockage Firebase :

- Les fichiers sont stockés dans Firebase Storage.
- Les métadonnées sont stockées dans Firestore.

En suivant ce guide, tu devrais être capable de mettre en place un flux complet pour traiter et stocker les fichiers 3D en utilisant Next.js et Django avec Firebase.