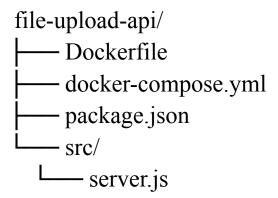
27. Mount any directory of host system to the container.

Folder Structure



1.Create a Folder

mkdir file-upload-api cd file-upload-api

2.create a Packtage.json

npm init -y
npm install express

3. Create src/server. js:

// src/server.js

```
const express = require('express');
const fs = require('fs');
const path = require('path');

const app = express();
const PORT = process.env.PORT || 3000;
const HOST DIR = process.env.HOST DIR || '/mounted';
```

```
// Endpoint to list files
app.get('/files', (req, res) => {
  fs.readdir(HOST_DIR, (err, files) => {
     if (err) {
       return res.status(500).json({ error: 'Unable to read directory'
});
     res.json(files);
  });
});
// Endpoint to get a specific file's contents
app.get('/files/:filename', (req, res) => {
  const filePath = path.join(HOST DIR, req.params.filename);
  fs.readFile(filePath, 'utf8', (err, data) => {
     if (err) {
       return res.status(404).json({ error: 'File not found' });
     res.send(data);
  });
});
app.listen(PORT, () => {
  console.log('File Viewer API running on
http://localhost:${PORT}`);
  console.log(`Serving files from ${HOST DIR}`);
});
```

4. Create Dockerfile

```
# Use an official Node.js image as the base
FROM node:18
# Set the working directory
WORKDIR /app
# Copy package.json and install dependencies
COPY package.json ./
RUN npm install
# Copy the rest of the application code
COPY src ./src
# Expose the application port
EXPOSE 3000
# Run the app
CMD ["node", "src/server.js"]
```

5. Create docker-compose.yml

```
version: '3'
services:
  file-upload:
  build: .
  ports:
    - '3000:3000'
  volumes:
    - ./uploads:/app/uploads
```

environment: - PORT=3000

6.Command

docker-compose up --build

For testing use a postman or thunder

To list files in the mounted directory:
http://localhost:3000/files

To view a specific file's content:
http://localhost:3000/files/<filename>

Use a get request and send any text file and show result

New directory is form /host-directory/

├── file1.txt

- file2.txt

└── file3.txt