OnlyOffice Document Integration Project

This project demonstrates how to create a React application that integrates with OnlyOffice Document Server for document editing and viewing.

Features

- File upload and management
- Document listing
- Integration with OnlyOffice Document Server
- Real-time document editing and collaboration
- File deletion

Project Structure

```
onlyoffice-integration/
├── client/                 # React frontend
│   ├── public/
│   ├── src/
│   │   ├── App.js          # Main application component
│   │   └── ...
│   ├── package.json
│   └── ...
├── server/                 # Node.js backend
│   ├── server.js           # Express server
│   └── package.json
├── files/                  # Storage for uploaded files
└── docker-compose.yml      # Docker Compose configuration
```

Setup Instructions

Prerequisites

- Node.js (v14 or later)
- npm or yarn
- Docker and Docker Compose (for containerized setup)

Option 1: Manual Setup

Backend Setup

1. Create the server directory and initialize the project:

```bash
mkdir -p onlyoffice-integration/server
cd onlyoffice-integration/server
npm init -y
```

2. Install the required dependencies:

```bash
npm install express express-fileupload cors axios
```

3. Create the `server.js` file with the provided code.
4. Create a directory for file storage:

```bash
mkdir -p ../files
```

Frontend Setup

1. Create a new React application:

```bash
cd ..
npx create-react-app client
cd client
```

2. Install required dependencies:

```bash
npm install lucide-react
```

3. Replace the content of `src/App.js` with the provided React component code.
4. Start the development server:

```bash
npm start
```

Option 2: Docker Compose Setup (Recommended)

1. Create project directories:

```bash
mkdir -p onlyoffice-integration/{client/src,server,files}
cd onlyoffice-integration
```

2. Create the necessary files:
   - Copy the React component code to `client/src/App.js`
   - Copy the Node.js server code to `server/server.js`
   - Create `client/package.json` and `server/package.json` with the necessary dependencies
   - Place the docker-compose.yml file in the root directory
3. Start the containers:

```bash
docker-compose up
```

This will start three containers:

- React frontend on http://localhost:3000
- Node.js backend on http://localhost:5000
- OnlyOffice Document Server on http://localhost:8080

OnlyOffice Document Server Configuration

The OnlyOffice Document Server is configured with JWT authentication for security. The JWT secret key is shared between the backend and document server for secure communication.

In a production environment, you should:

1. Change the JWT secret to a strong random string
2. Set up HTTPS for all services
3. Configure proper document storage (like cloud storage, database, etc.)
4. Implement proper user authentication and authorization

Customization

Supported File Types

The application supports the following file types:

- Word documents (.doc, .docx, .txt, .rtf)
- Spreadsheets (.xls, .xlsx, .csv)
- Presentations (.ppt, .pptx)
- PDF documents (.pdf)

To add support for additional file types, modify the `getFileType` function in both the frontend and backend code.

Styling

The frontend uses Tailwind CSS-like utility classes for styling. You can customize the appearance by modifying these classes in the React components.

Production Deployment

For production deployment:

1. Build the React application:

   ```bash
   cd client
   npm run build
   ```

2. Configure the backend to serve the static files from the build directory
3. Set up a proper database for file metadata storage
4. Configure proper storage for uploaded files (like AWS S3, Google Cloud Storage, etc.)
5. Set up a reverse proxy (like Nginx) to serve the application
6. Configure HTTPS for all services

Troubleshooting

Common Issues

1. **CORS errors**: Make sure the backend is properly configured to allow cross-origin requests from the frontend.
2. **File upload issues**: Check file size limits and permissions on the storage directory.
3. **OnlyOffice Document Server connection issues**: Verify the document server is running and accessible at the configured URL.

OnlyOffice Document Server Logs

To check the OnlyOffice Document Server logs:

```bash
docker logs onlyoffice-document-server
```

Resources

- OnlyOffice Document Server API Documentation
- OnlyOffice Document Server Official Docker Image
- Express.js Documentation
- React Documentation