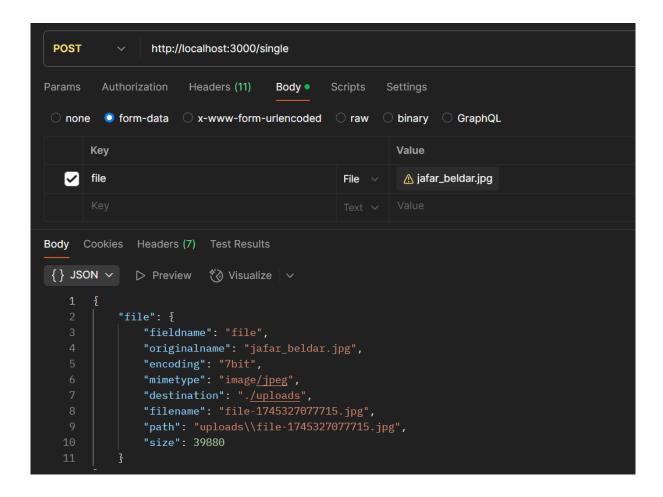
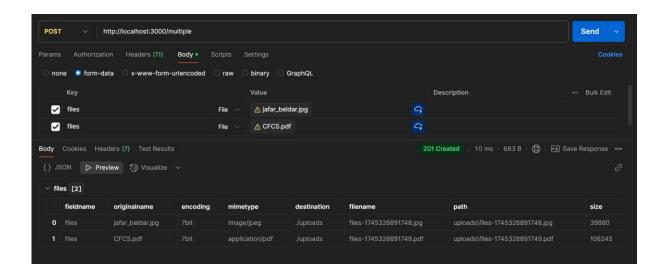
## Day14 - Multer, Buffer and File upload

## **Multer: Single and Multiple Files**

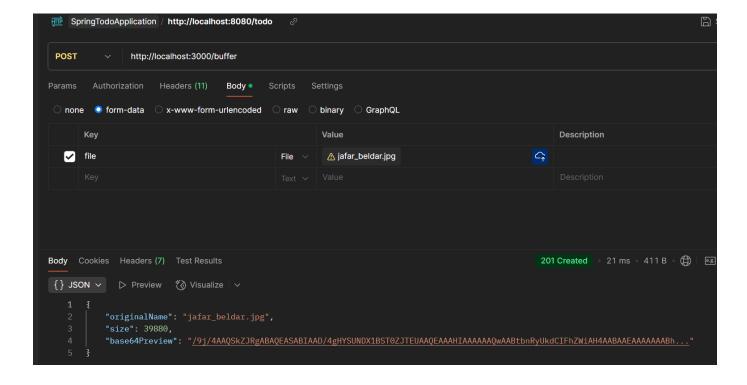
```
import { AppService } from './app.service';
3 import { Controller, Post, Get, UploadedFile, UploadedFiles,UseInterceptors,} from '@nestjs/common';
4 import { FileInterceptor, FilesInterceptor } from '@nestjs/platform-express';
5 import { diskStorage } from 'multer';
6 import { extname } from 'path';
8  @Controller()
   export class AppController {
     constructor(private readonly appService: AppService) {}
     @Get()
     getHello(): string {
      return this.appService.getHello();
     @Post('single')
     @UseInterceptors(FileInterceptor('file', {
       storage: diskStorage({
          destination: './uploads',
          filename: (req, file, callback) => {
           const unique = Date.now() + extname(file.originalname);
           callback(null, `${file.fieldname}-${unique}`);
       }),
     uploadSingle(@UploadedFile() file: Express.Multer.File) {
       return { file };
      @Post('multiple')
     @UseInterceptors(FilesInterceptor('files', 5, {
       storage: diskStorage({
          destination: './uploads',
          filename: (req, file, callback) => {
           const unique = Date.now() + extname(file.originalname);
           callback(null, `${file.fieldname}-${unique}`);
         },
     uploadMultiple(@UploadedFiles() files: Express.Multer.File[]) {
       return { files };
```





## **Buffer:**

```
1 @Post('buffer')
2 @UseInterceptors(FileInterceptor('file'))
3 handleBuffer(@UploadedFile() file: Express.Multer.File) {
4    const base64 = file.buffer.toString('base64');
5    return {
6     originalName: file.originalname,
7     size: file.size,
8     base64Preview: base64.slice(0, 100) + '...',
9    };
10 }
```



Iafar Beldar