Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: «ВТ»

Тема: “Работа с файлами”

Выполнил:

Студент 3 курса

Группы ПО-7

Комиссаров А.Е.

Проверил:

Михняев А.Л.

2022

**Цель работы:** Изучить возможности работы с файлами и каталогами в веб-среде*.*

1. Написать скрипт, выводящий содержимое корня веб-сервера.
2. Написать скрипт создания каталога на диске.
3. Написать скрипт переименования каталога.
4. Написать скрипт удаления каталога.
5. Написать скрипт для отправки файлов на сервер. Размер сохраняемого на сервер файла не должен превышать 100 Kb. После отправки пользователь должен получить информацию о размере переданного файла, был ли файл перезаписан, был ли файл присоединен к другому файлу, был ли файл сохранен на сервере, директорию файла, имя файла, его расширение, время создания и последней модификации.
6. Написать скрипт переименования файла.
7. Написать скрипт копирования файла.
8. Написать скрипт удаления файла.
9. Написать скрипт, считывающий содержимое текстового файла и выводящий это содержимое на экран.
10. Написать функционал, передающий из html - формы присоединённый файл расширений .jpg или .gif на сервер. На сервере принятый файл сохраняется в базу данных, потом считывается оттуда и выводится на экран.

**Ход работы.**

@Injectable()

export class FilesService {

constructor(private readonly configService: ConfigService, private readonly prisma: PrismaService) {}

async getAllFiles(params: { where?: Prisma.FileWhereInput }): Promise<File[]> {

const { where } = params

return this.prisma.file.findMany({

where,

})

}

async readDirectory(): Promise<string[]> {

return readdirSync(uploadsPath).map(file => file)

}

async createFolder(folder: string): Promise<string> {

const newFolder = uploadsPath + folder

if (existsSync(newFolder)) {

throw new BadRequestException("Such folder is already exists")

}

await ensureDir(newFolder)

return folder

}

async rename(dto: RenameDto): Promise<string> {

const { name, newName, folder } = dto

let newPath\_ = uploadsPath

let path\_ = uploadsPath

if (folder) {

newPath\_ += folder + "/" + newName

path\_ += folder + "/" + name

} else {

newPath\_ += newName

path\_ += name

}

if (existsSync(newPath\_)) {

throw new BadRequestException("Such file or directory is already exists")

}

if (!existsSync(path\_)) {

throw new BadRequestException("There is no such file or directory")

}

if (lstatSync(path\_).isDirectory()) {

copySync(path\_, newPath\_, { recursive: true })

rmSync(path\_, { recursive: true })

await this.prisma.file.updateMany({

where: {

folder: name,

},

data: {

folder: newName,

},

})

} else if (lstatSync(path\_).isFile()) {

await this.prisma.file.update({

where: {

name\_folder: {

name,

folder,

},

},

data: {

url: this.configService.get("SERVER\_URL") + "/uploads/" + folder + "/" + newName,

name: newName,

},

})

renameSync(path\_, newPath\_)

}

return newName

}

async deleteFolder(name: string): Promise<string> {

if (!existsSync(uploadsPath + name)) {

throw new BadRequestException("Such folder doesn't exists")

}

rmSync(uploadsPath + name, { recursive: true })

await this.prisma.file.deleteMany({

where: {

folder: name,

},

})

return name

}

async uploadFile(dto: UploadFileDto): Promise<File> {

try {

const { file, folder } = dto

const uploadFolder = uploadsPath + folder

if (!existsSync(uploadFolder)) {

throw new BadRequestException("There is no such folder")

}

const serverUrl = this.configService.get("SERVER\_URL")

await writeFile(`${uploadFolder}/${file.originalname}`, file.buffer)

return this.prisma.file.create({

data: {

name: file.originalname,

url: `${serverUrl}/uploads/${folder}/${file.originalname}`,

size: file.size,

mimetype: file.mimetype,

folder,

},

})

} catch (error) {

throw new HttpException("Error while creating file", HttpStatus.INTERNAL\_SERVER\_ERROR)

}

}

async copyFile(dto: CopyFileDto): Promise<File> {

const src = uploadsPath + dto.from

const dest = uploadsPath + dto.to

if (!existsSync(src)) {

throw new BadRequestException("Such file doesn't exists")

}

const from\_name\_folder = dto.from.split("/")

const to\_name\_folder = dto.to.split("/")

const file = await this.prisma.file.findUnique({

where: {

name\_folder: {

folder: from\_name\_folder[0],

name: from\_name\_folder[1],

},

},

})

copySync(src, dest)

return this.prisma.file.create({

data: {

url: file.url,

size: file.size,

mimetype: file.mimetype,

folder: to\_name\_folder[0],

name: to\_name\_folder[1],

},

})

}

async deleteFile(dto: DeleteFileDto): Promise<File> {

const deletePath = uploadsPath + dto.folder + "/" + dto.name

if (!existsSync(deletePath)) {

throw new BadRequestException("Such file doesn't exists")

}

rmSync(deletePath)

return this.prisma.file.delete({

where: {

name\_folder: {

name: dto.name,

folder: dto.folder,

},

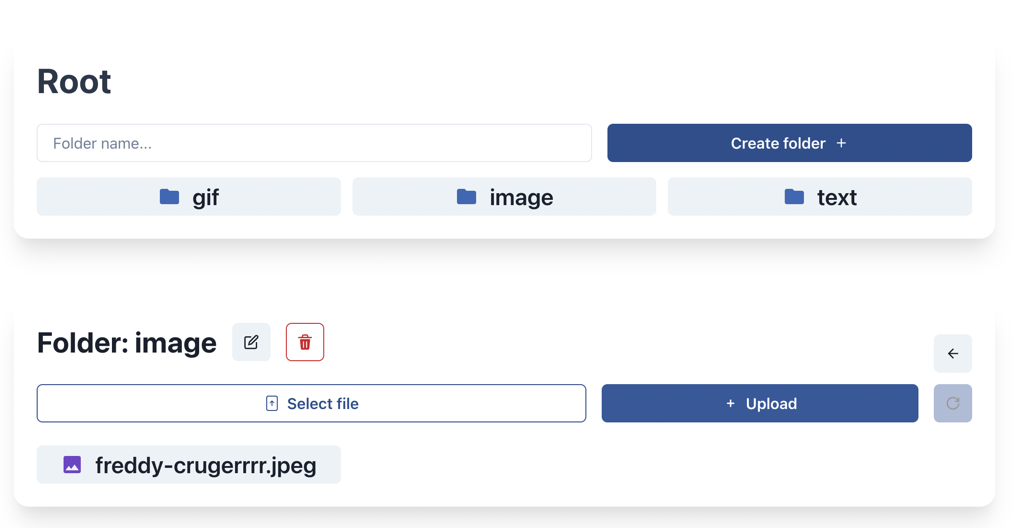
},

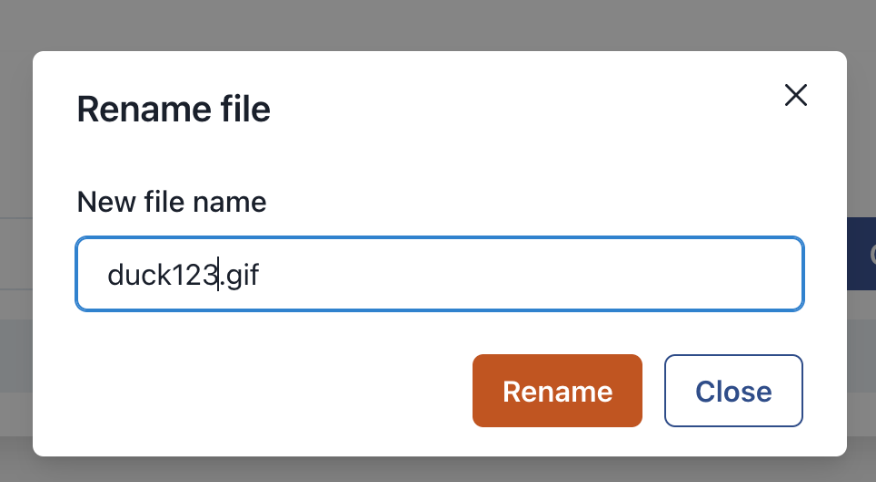
})

}

}

Результаты выполнения программы:





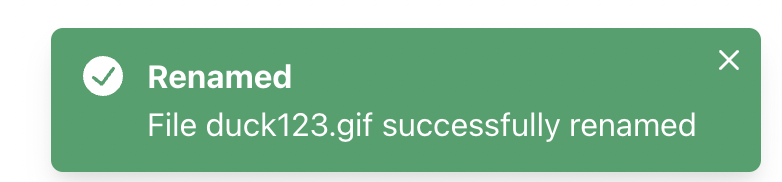


Рис. 1,2,3 – создание, переименовывание файла.

**Вывод:** Я изучил возможности работы с файлами и каталогами в веб-среде.