

Đồ án công nghệ Phần mềm

PROOF OF CONCEPT

GVHD: Thầy Ngô Huy Biên

Nhóm 01

Vấn đề:

- Request: File
- Response: fileName và URL dẫn tới file
- Lưu trữ và quản lý file trong bucket

Hướng giải quyết

- Build 1 con CDN chạy độc lập đóng vai trò lưu trữ file và trả về liên kết tới file đó, không đóng vai trò quản lý bucket hay xử lý logic
- Project có 3 repo độc lập bao gồm frontend, backend, CDN-service

cdn-service Private

0 0 0 0 Updated 2 hours ago

K3-backend Private

Back end for K3 - Simple Storage Service

docker microservice event-sourcing ddd-architecture nestjs

K3-frontend Private

Front end for K3 - Simple Storage Service

JavaScript GPL-3.0 0 0 0 0 Updated on May 12

Code mẫu FrontEnd (base64)

```
const file = await urlToFile(imageAfterChange, "avatar.jpg", "image/jpg");
var data = new FormData();
data.append("files", file);

// now upload
const config = {
  headers: {
    authorization: localStorage.getItem("token"),
    "Content-Type": "application/x-www-form-urlencoded",
  },
};
axios.post(FILE_URL, data, config).then((response) => {
  // console.log(response);
  setUrl(response[0].url);
});
} else {
  return;
}
```

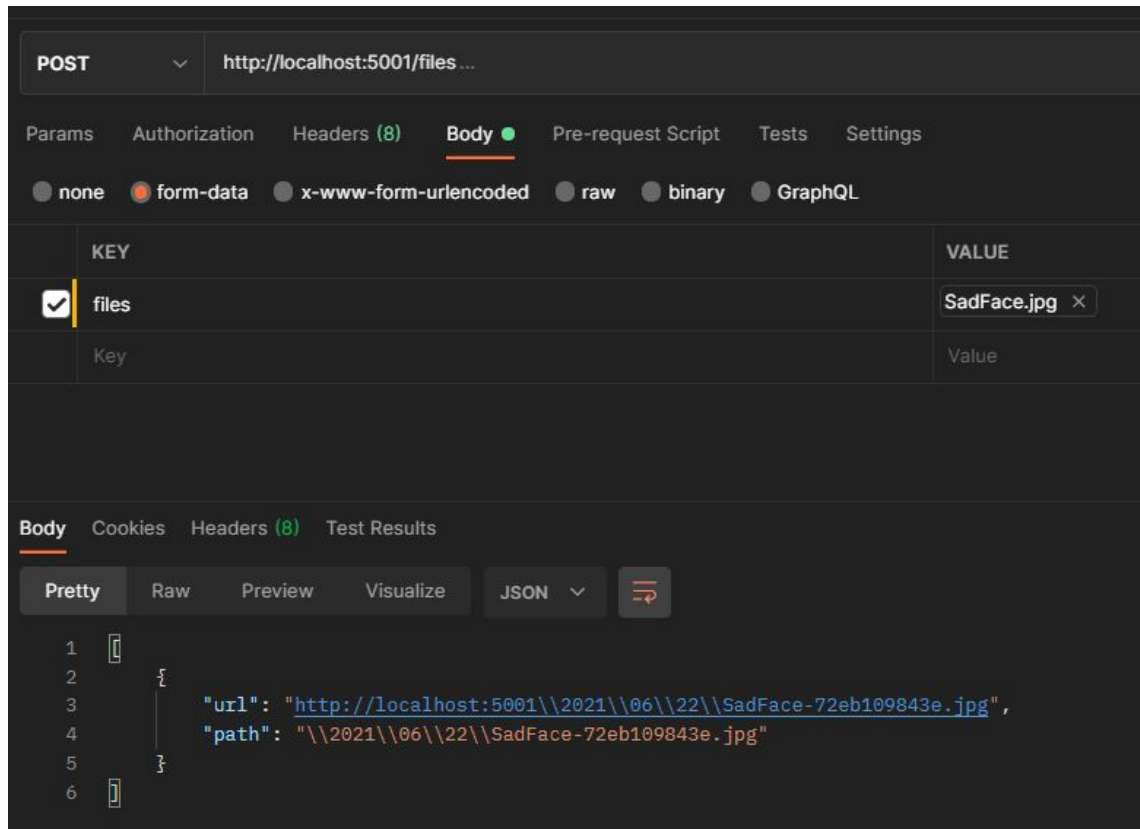
Code mẫu (CDN - Service)

Render FileName and FilePath

```
exports.mediaFileFilter = mediaFileFilter;
const editFileName = (req, file, callback) => {
  const name = file.originalname.split('.')[0].replace(/\s+/g, '');
  const fileExtName = path_1.extname(file.originalname);
  const randomName = Array(10)
    .fill(null)
    .map(() => Math.round(Math.random() * 16).toString(16))
    .join('');
  callback(null, `${name}-${randomName}${fileExtName}`);
};
export const editFilePath = (req: any, file: any, callback: any) => Promise<void>
const editFilePath = async (req, file, callback) => {
  const dateNow = Date.now();
  const date = moment(dateNow).format('YYYY/MM/DD');
  const filePath = `./media/${date}`;
  console.log('filepath:', filePath);
  console.log(path_1.join(__dirname, '../..../'));
  if (!fs.existsSync(filePath)) {
    const year = moment(dateNow).format('YYYY');
    const month = moment(dateNow).format('MM');
    const day = moment(dateNow).format('DD');
    await (function () {
      if (!fs.existsSync(`./media/${year}`)) {
        fs.mkdirSync(`./media/${year}`);
      }
    })();
    await (function () {
      if (!fs.existsSync(`./media/${year}/${month}`)) {
        fs.mkdirSync(`./media/${year}/${month}`);
      }
    })();
    await (function () {
      if (!fs.existsSync(`./media/${year}/${month}/${day}`)) {
        fs.mkdirSync(`./media/${year}/${month}/${day}`);
      }
    })();
  }
  callback(null, filePath);
};
exports.editFilePath = editFilePath;
```

Kết quả thu được

Sử dụng Postman



Code mẫu (CDN - Service)

Controler

```
@Controller()
export class FileController {
  @Post('/files')
  @UseInterceptors(
    FilesInterceptor('files', 20, {
      storage: diskStorage({
        destination: editFilePath,
        filename: editFileName,
      }),
      fileFilter: mediaFileFilter,
    })
  )
  uploadMultipleFiles(@UploadedFiles() files) {
    const response = [];
    files.forEach((file) => {
      const fileReponse = {
        url: 'http://localhost:5001' + file.path.replace('media', ''),
        path: file.path.replace('media', ''),
      };
      console.log('File:', file);
      response.push(fileReponse);
    });
    return response;
  }

  @Get('/:year/:month/:day/:fileName')
  getFile(
    @Param('year') year,
    @Param('month') month,
    @Param('day') day,
    @Param('fileName') fileName,
    @Res() res,
  ) {
    return res.sendFile(fileName, { root: `./media/${year}/${month}/${day}` });
  }
}
```