

CENTRALMUSIC - LDS
TESTES DE GESTÃO DE IMAGENS

Version 2.0

18/02/2021

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	<i>Marcelo Carvalho</i>	<i>18/02/2021</i>	<i>Marcelo Carvalho</i>	<i>18/02/2021</i>	Criação do documento e adaptação para o projeto. Registo de testes relativos às imagens
2.0	<i>Samuel Cunha</i>	<i>18/02/2021</i>	<i>Samuel Cunha</i>	<i>18/02/2021</i>	Criação do documento e adaptação para o projeto. Registo de testes relativos às imagens

Índice de Figuras

Figura 1 Diretório de imagens.....	2
Figura 2 UpdateImagePublication.....	3
Figura 3 DeleteImage	4
Figura 4 UploadProfilePic.....	5
Figura 5 DeleteProfilePicture	6

Índice

1.	Introdução.....	1
1.1	Identificador do documento	1
1.2	Âmbito.....	1
2.	Métodos a testar.....	2
2.1	Descrição dos métodos	2
2.2	Estrutura do diretório das imagens	2
2.3	Testes	3
2.3.1	UpdateImagePublication	3
2.3.2	DeleteImage	4
2.3.3	UploadProfilePic.....	5
2.3.4	DeleteProfilePicture	6
2.3.5	Métodos para visualizar Imagens guardadas no servidor	7
3.	Conclusão	7

1. Introdução

1.1 Identificador do documento

TGI18022021V2.0

Descrição do identificador do projeto:

TDS – TestDesignSpecification

18 - Número do mês (dois dígitos seguidos dos dois dígitos iniciais)

02 - Dia do mês (dois dígitos iniciais)

2021 - Ano de criação (quatro dígitos finais)

V2.0 - Número de versão do projeto (dois caracteres sendo “V” o identificador de versão através de sigla seguido do número de versão “2” e número de alteração “.0”)

1.2 Âmbito

Este documento serve como suplemento aos testes, de modo a referir dos testes relativos às imagens da aplicação CentralMusic. Este documento foi desenvolvido com o propósito de explicar como foram testadas as funcionalidades relativas ao tratamento das imagens dos utilizadores e das Publications, visto que durante o tempo de desenvolvimento do trabalho não foi encontrada maneira de realizar *unit tests* para a gestão das imagens.

Nas secções seguintes, será feita a descrição do âmbito onde se enquadram os testes, os métodos relativos a gestão de imagens e serão apresentados os respetivos resultados.

Os testes enquadram-se na aplicação CentralMusic, constituída por um componente backend desenvolvido em dotNet 5 e um componente frontend, desenvolvido em Flutter, no âmbito da disciplina de Laboratório de Desenvolvimento de Software. O produto a ser testado é uma aplicação (web e mobile), nomeadamente a funcionalidade de gestão de imagens, com o objetivo de permitir aos utilizadores mostrarem os artigos nas publicações e também terem uma imagem de perfil.

2. Métodos a testar

2.1 Descrição dos métodos

Os métodos sujeitos a testes pertencem a classe *ImageUploadHelper.cs*, responsável *por* fazer a gestão de Imagens, mais precisamente:

- ✓ UpdateImagePublication – Faz upload de uma coleção de imagens para uma Publicação.
- ✓ DeleteImage Apaga uma imagem que pertence a uma *Publicação*.
- ✓ UploadProfilePic – Faz upload de uma imagem para um utilizador.
- ✓ DeleteProfilePicture – Apaga uma imagem que pertence a um utilizador.

2.2 Estrutura do diretório das imagens

As imagens são guardadas na pasta “wwwroot\\images”, separadas por “Posts” para imagens de *Posts*, e “Users” para imagens de utilizadores, porém imagem de destaque de uma Publicação é guardada numa pasta “main”. Como podemos observar na figura seguinte:

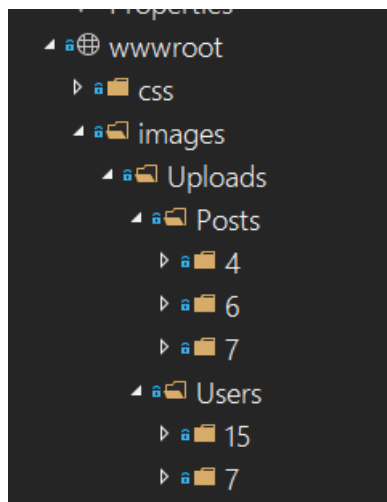


Figura 1 Diretório de imagens

2.3 Testes

Os testes foram realizados usando a ferramenta Swagger. Nesta secção serão mostradas imagens relativas aos testes e o resultado obtido. Todas as imagens são as correspondentes às da [Figura 1 Diretório de imagens](#). A publicação usada tem Id igual a 6, e o utilizador Id igual a 15.

2.3.1 UpdateImagePublication

Método para fazer upload de imagens para uma *Publicação*. Para fazer o upload de imagens para uma *Publicação*, utiliza-se a rota:

`https://localhost:5001/api/Publication/UpdateImagePublication`, em que 6 é o Id da Publicação. O resultado consta na figura seguinte:

Name	Description
id integer(\$int32) <small>(query)</small>	<input type="text" value="6"/>

Request body

multipart/form-data ▾

images
array

2021-01-05 (1).png

+

Add item

☐ Send empty value

mainImage
string(binary)

2021-01-05.png

☐ Send empty value

Execute

Responses

```
Curl
curl -X POST "https://localhost:5001/api/Publication/UpdateImagePublication?id=6" -H "accept: /*/*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyogZXRlcC2VyaWwmbnVtYmwyOjoiMTUxLlQ3bmFpbCEiaGlhcmlibG8vQGdtYWlsLnkvbnVtYmwyOjE2MTRMNkAASOTisInW4cCI6MTYxeHZYMNDUSKswlaWF0IjozdGFjezNjcuc0TkyfQ.V2uxgFlmSpY48lnjtvtIncye9CpnVQSuvPkDg7g0yMc" -H "Content-Type: multipart/form-data" -F "Images=@2021-01-05 (1).png;type=image/png" -F "mainImage=@2021-01-05.png;type=image/png"
```

Request URL

https://localhost:5001/api/Publication/UpdateImagePublication?id=6

Server response

Code	Details
200	Response body <div>true</div>

Figura 2 UpdateImagePublication

Este método recorre ao método `UpdateImagePublication` para fazer o upload de um array de imagens de imagens e uma imagem de destaque.

2.3.2 DeletelImage

Método para apagar uma imagem de um *Publicação*. Para apagar uma imagem de uma *Publicação*, utiliza-se a rota:

`https://localhost:5001/api/Publication/deletelImage/{postId}`

O resultado consta na figura seguinte:

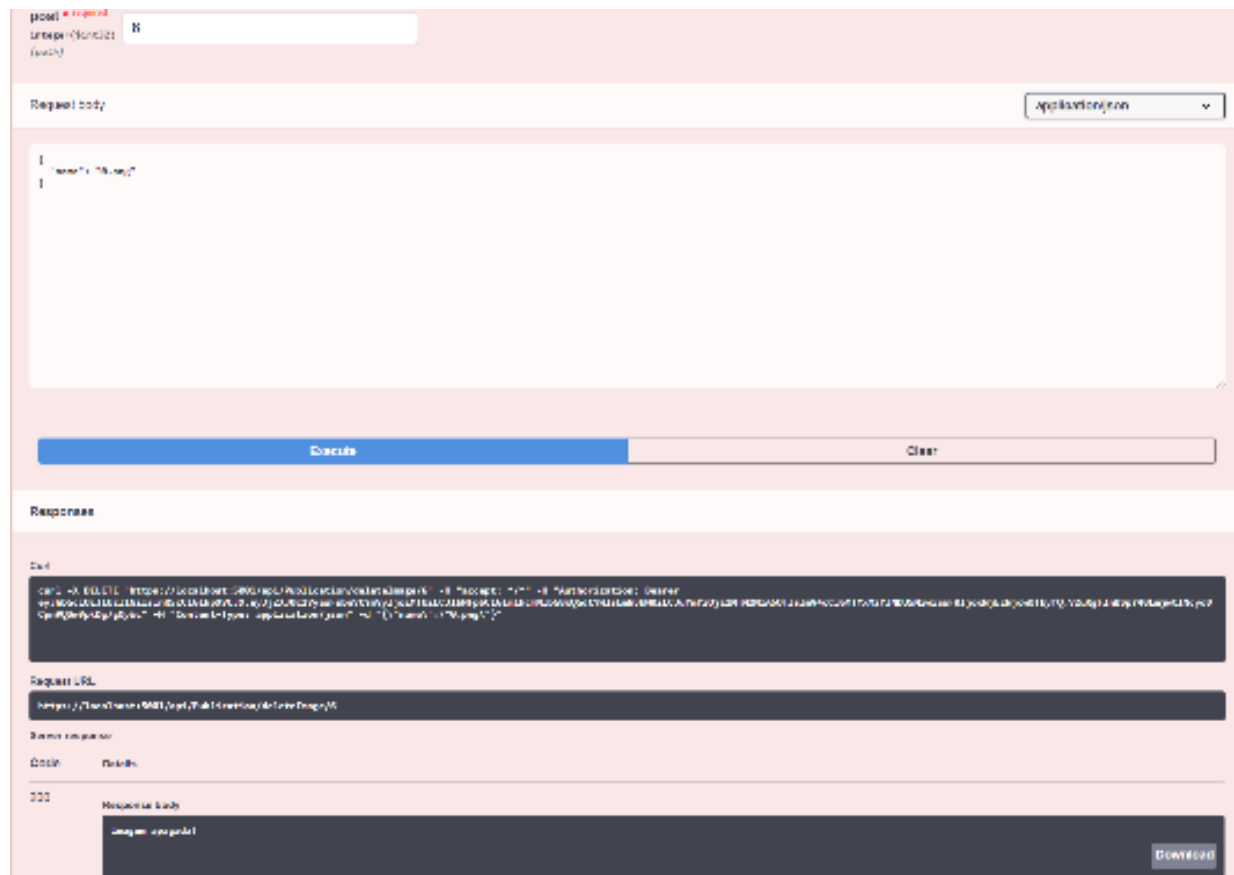


Figura 3 DeletelImage

2.3.3 UploadProfilePic

Método para fazer o upload da imagem de um Utilizador, através da seguinte rota:

https://localhost:5001 /api/User/UploadProfilePic

O resultado consta na figura seguinte:

Request body

multipart/form-data ▼

profilePic
string(\$binary)

E Escolher arquivo Object em JS.png

☐ Send empty value

ExecuteClear

Responses

Curl

```
curl -X POST "https://localhost:5001/api/User/UploadProfilePic" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3ZXRlc2VyaWFnbnVtYmVyIjoiaWNTUiLCJlbWFpbCI6Im1hcmlkSGxQdGVlLnMvbm81LjcuYmYyOjE2MTk2NzAsOTIwInV4Ci6MTTYxhzY3NDUSMiwiYWFi0TjoxNjEzMjc0OTkyfQ.V2uXgFlnBSspY48LmjwtINcyo9CpnVQSwpkdD7gOylmc" -H "Content-Type: multipart/form-data" -F "profilePic=@Object em JS.png;type=image/png"
```

Request URL

https://localhost:5001/api/User/UploadProfilePic

Server response

Code	Details
200	<div><div>Response body<div>trueDownload</div></div><div><div>Response headers<div>access-control-allow-origin: * content-type: application/json; charset=utf-8 date: Thu18 Feb 2021 18:15:27 GMT server: Kestrel</div></div></div></div>

Responses

Code	Description	Links
200	Success	No links

Figura 4 UploadProfilePic

2.3.4 DeleteProfilePicture

Método para fazer eliminar a imagem de um Utilizador, através da seguinte rota:

[https://localhost:5001 /api/User/deleteProfilePicture](https://localhost:5001/api/User/deleteProfilePicture)

O resultado consta na figura seguinte:

DELETE /api/User/deleteProfilePicture

Parameters

Cancel

No parameters

ExecuteClear

Responses

Curl

```
curl -X DELETE "https://localhost:5001/api/User/deleteProfilePicture" -H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZXJ8c2VyaWFsbnVtYmVyIjo1MTU5LCJ1bWFnFpbCI6ImlhcmNlbG8xQGdtYm1sLmNvbW8iLCJ1bWYiOiJlE2MTM2NzA5OTIsImV4cCI6MTYxMzY3NDU5MiwiaWF0IjoxNjEzNjcwOTkyfQ.V2uXgF1nBSpY48LmjvtINcyo9CpnVQSwVpkDg7g0yWc"
```

Request URL

```
https://localhost:5001/api/User/deleteProfilePicture
```

Server response

Code	Details
200	<div>Response body<pre>{ "success": "Imagem apagada!"}</pre><div>Download</div>Response headers<pre>access-control-allow-origin: * content-type: application/json; charset=utf-8 date: Thu18 Feb 2021 18:16:27 GMT server: Kestrel</pre></div>

Responses

Code	Description	Links
200	Success	No links

Figura 5 DeleteProfilePicture

2.3.5 Métodos para visualizar Imagens guardadas no servidor

É possível pedir o caminho específico de uma imagem guardado na base de dados mas o motivo de guardar as imagens na pasta “wwwroot” (pasta de ficheiros estáticos do servidor) é para que seja possível aceder as mesmas através de um link direto, sendo que todas as imagens são publicas não existe qualquer problema em manter o acesso aberto.

3. Conclusão

Dados os testes concluídos, deu-se como concluído o desenvolvimento da funcionalidade. A devida utilização da pasta estática do servidor mostra como é possível a interação do servidor com ficheiros e manter a performance do servidor sem ter de comprometer o mesmo enviar ficheiros byte by byte.