
Distributed Computing

P02_WSPProject

— —

Pau Francino
Joel Romia

Index

- Classes and UML
- Endpoint, operations, resulting codes,...
- Execution cases and outcomes
- Project Information
- Execution Demo

Classes and UML

**Endpoints, operations, resulting
codes,...**



Representation

Files:

- Get All Files
- Get File By Hash
- Get File Title by Hash
- Get File Description by Hash
- Get File Keyword by Hash
- Get File Title by Id
- Get File Description by Id
- Get File Keyword by Id

Search:

- Anytype
- Title
- Description
- Keyword

Post:

- addFile
- addFileTitle
- addFileDescription
- addFileKeyword

Get All Files (Files)

Get /files

Get all files in repository

Path Parameters

Empty

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File By Hash (Files)

Get /files/{hash}

Get all content from a specific file

Path Parameters

hash-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Title By Hash (Files)

Get /files/{hash}/titles

Get content from files by content type titles

Path Parameters

hash-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Description By Hash (Files)

Get /files/{hash}/descriptions

Get content from files by content type descriptions

Path Parameters

hash-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Keyword By Hash (Files)

Get /files/{hash}/keywords

Get content from files by content type keywords

Path Parameters

hash-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Title By Id (Files)

Get /files/{hash}/titles/{id}

Get content from files by content id

Path Parameters

hash-> String-> required

id-> String -> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Description By Id (Files)

Get /files/{hash}/description/{id}

Get content from files by content id

Path Parameters

hash-> String-> required

id-> String -> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Get File Keyword By Id (Files)

Get /files/{hash}/keyword/{id}

Get content from files by content id

Path Parameters

hash-> String-> required

id-> String -> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Search By Anytype (Search)

Get /files/search

Search by any content

Path Parameters

anytype-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Search By Title(Search)

Get /files/search/title

Search by title

Path Parameters

title-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Search By Description(Search)

Get /files/search/description

Search by description

Path Parameters

descr-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Search By Keyword(Search)

Get /files/search/keyword

Search by keyword

Path Parameters

keyw-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.OK

Add File (Post)

Post /files

Post a file

Path Parameters

hash-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.CREATED

Add File Title (Post)

Post /files/{hash}/title

Post a file title

Path Parameters

hash-> String-> required

title-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.CREATED

400 -> HttpStatus.BAD_REQUEST

Add File Description (Post)

Post /files/{hash}/description

Post a file description

Path Parameters

hash-> String-> required

description-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.CREATED

400 -> HttpStatus.BAD_REQUEST

Add File Keyword (Post)

Post /files/{hash}/keyword

Post a file keyword

Path Parameters

hash-> String-> required

keyword-> String-> required

Query String

Empty

Responses

200 -> HttpStatus.CREATED

400 -> HttpStatus.BAD_REQUEST

Execution cases and Outcome

Get All Files (Files)

GET

localhost:8080/files

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

20 ms

397 B

Pretty

Raw

Preview

Visualize

Text

1

Content for file with hash: hash4_1

2

Titles: Cocina

3

Descriptions: Manual de cocina

4

Keywords: cocina, aprender

5

Content for file with hash: hash5_1

6

Titles: series

7

Descriptions: New series in Netflix

8

Keywords: Series Netflix

9

Get File By Hash

GET

localhost:8080/files/hash4_1

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK13 ms279 B

Pretty

Raw

Preview

Visualize

Text

1

Content for file with hash: hash4_1

2

Titles: Cocina

3

Descriptions: Manual de cocina

4

Keywords: cocina, aprender

5

Get File Title By Hash

GET

localhost:8080/files/hash4_1/titles

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

10 ms

215 B

Pretty

Raw

Preview

Visualize

Text

1

Titles for file with hash: hash4_1

2

Titles: Cocina

3

Get File Description By Hash

GET

localhost:8080/files/hash4_1/descriptions

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK6 ms238 B

Pretty

Raw

Preview

Visualize

Text

1

Descriptions for file with hash: hash4_1

2

Descriptions: Manual de cocina

3

Get File Keyword By Hash

GET

localhost:8080/files/hash4_1/keywords

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

7 ms

230 B

Pretty

Raw

Preview

Visualize

Text

1

Keywords for file with hash: hash4_1

2

Keywords: cocina, aprender

3

Get File Title By Id

GET

localhost:8080/files/hash4_1/titles/0

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

16 ms

169 B

Pretty

Raw

Preview

Visualize

Text

1

Cocina

Get File Description By Id

GET

localhost:8080/files/hash4_1/description/0

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

8 ms

180 B

Pretty

Raw

Preview

Visualize

Text

1 Manual de cocina

Get File Keyword By Id (Files)

GET

localhost:8080/files/hash4_1/keyword/0

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

200 OK

9 ms

180 B

Pretty

Raw

Preview

Visualize

Text

1

cocina, aprender

Search By Anytype (Search)

GET

localhost:8080/files/search

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Cocina

Body

Cookies

Headers (5)

Test Results

200 OK

13 ms

304 B

Pretty

Raw

Preview

Visualize

Text

1

Found files with string Cocina as title: Cocina

2

Found files with string Cocina as description:

3

Found files with string Cocina as keyword:

4

Search By Title(Search)

GET

localhost:8080/files/search/title

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Cocina

Body

Cookies

Headers (5)

Test Results

200 OK

6 ms

213 B

Pretty

Raw

Preview

Visualize

Text

1

Found files with string Cocina as title: Cocina

2

Search By Description(Search)

GET

localhost:8080/files/description

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Manual de cocina

Body

Cookies

Headers (5)

Test Results

200 OK

8 ms

239 B

Pretty

Raw

Preview

Visualize

Text

1

Found files with string Manual de cocina as description: Manual de cocina

2

Search By Keyword(Search)

GET

localhost:8080/files/keyword

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1 Series

Body

Cookies

Headers (5)

Test Results

200 OK

14 ms

215 B

Pretty

Raw

Preview

Visualize

Text

1 Found files with string Series as keyword: Series

2

Add File (Post)

POST

localhost:8080/files

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

hash6_1

Body

Cookies

Headers (5)

Test Results

201 Created

13 ms

181 B

Pretty

Raw

Preview

Visualize

Text

1

File created

Add File Title (Post)

POST

localhost:8080/files/hash5_1_/title

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Naruto

Body

Cookies

Headers (5)

Test Results

201 Created

9 ms

180 B

Pretty

Raw

Preview

Visualize

Text

1

Title added

Add File Description (Post)

POST

localhost:8080/files/hash5_1_/description

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Best cartoon serie in HBO

Body

Cookies

Headers (5)

Test Results

201 Created

14 ms

186 B

Pretty

Raw

Preview

Visualize

Text

1

Description added

Add File Keyword (Post)

POST

localhost:8080/files/hash5_1_/keyword

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Cartoon

Body

Cookies

Headers (5)

Test Results

201 Created

10 ms

182 B

Pretty

Raw

Preview

Visualize

Text

1

Keyword added

Project Information

Github link: https://github.com/romi99fr/MultimediaInfo_WebService

Hours: 70-90

Execution Demo

1- Create four different digital contents, the first and the second with similar description (e.g., First episode of the wire, first episode of the wire HBO tv show)

2- Modify 3rd's description

3- Search a content by its description (full)

4- Delete the fourth one

5- List all contents

~~6- Create two different users~~

7- Upload a different content to each previous user

8- Delete the second content uploaded

9- Search content by its description (partial)

~~10- Show its user~~

~~11- Check some ID from an existent user~~

~~12- Check some other nonexistent ID~~