Distributed Computing P02_WSProject

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

Delete:

- All Files
- A File
- File Tiltes
- File Descriptions
- File Keywords

Get All Files (Files)

Get /files

Get all files in repository

Path Parameters

Empty

Query String

Empty

Responses

Get File By Hash (Files)

Get /files/{hash}

Get all content from a specific file

Path Parameters

hash-> String-> required

Query String

Empty

Responses

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

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

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

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

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

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

Search By Anytype (Search)

Get /files/search

Search by any content

<u>Path Parameters</u>

anytype-> String-> required

Query String

Empty

Responses

Search By Title(Search)

Get /files/search/title

Search by title

<u>Path Parameters</u> title-> String-> required

Empty

Query String

Responses

Search By Description(Search)

Get /files/search/description

Search by description

<u>Path Parameters</u>

descr-> String-> required

Query String

Empty

Responses

Search By Keyword(Search)

Get /files/search/keyword

Search by keyword

Path Parameters

keyw-> String-> required

Query String

Empty

Responses

Add File (Post)

Post /files

Post a file

Path Parameters

hash-> String-> required

Query String

Empty

Responses

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

Delete All Files (Delete)

Delete /files

Delete all Files

Path Parameters

hash-> String-> required

Query String Responses

Empty

Delete a File (Delete)

Delete /files/{hash}

Delete a File by its hash

Path Parameters

hash-> String-> required

Query String Responses

Empty

Delete File Titles (Delete)

Delete /files/{hash}/titles

Delete all File titles by its hash

Path Parameters

hash-> String-> required

Query String Responses

Empty

Delete File Descriptions (Delete)

Delete /files/{hash}/descriptions

Delete all File descriptions by its hash

Path Parameters

hash-> String-> required

Query String Responses

Empty

Delete File Keywords (Delete)

Delete /files/{hash}/keywords

Delete all File keywords by its hash

Path Parameters

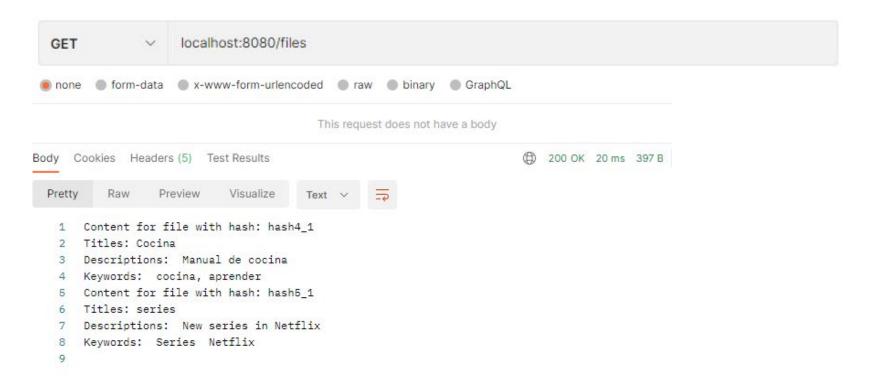
hash-> String-> required

Query String Responses

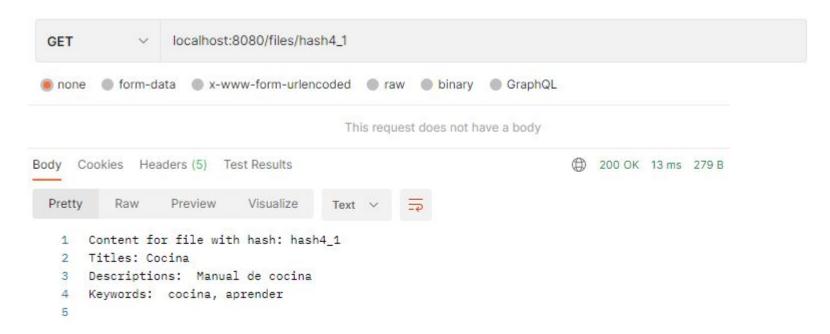
Empty

Execution cases and Outcome

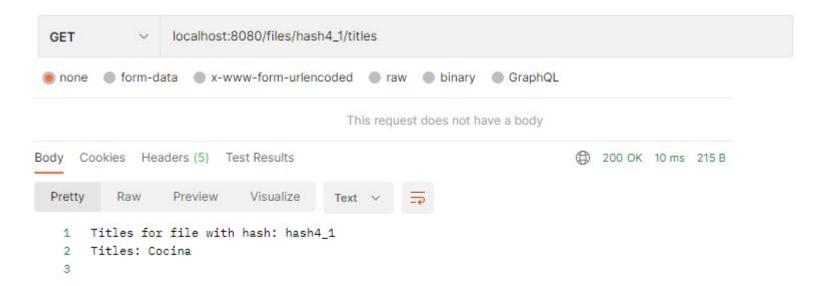
Get All Files (Files)



Get File By Hash



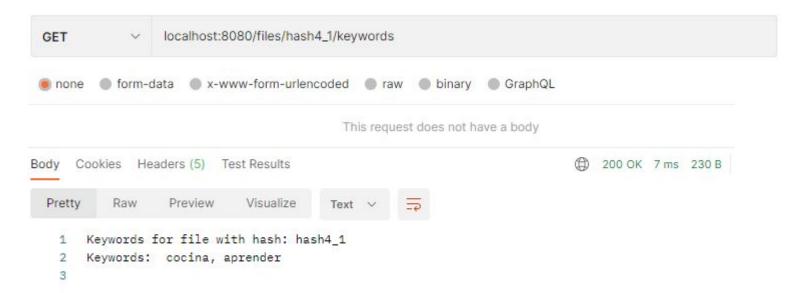
Get File Title By Hash



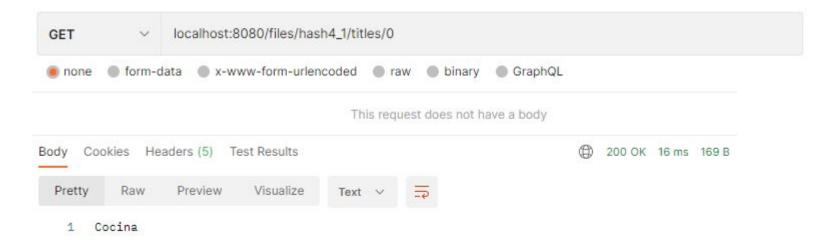
Get File Description By Hash



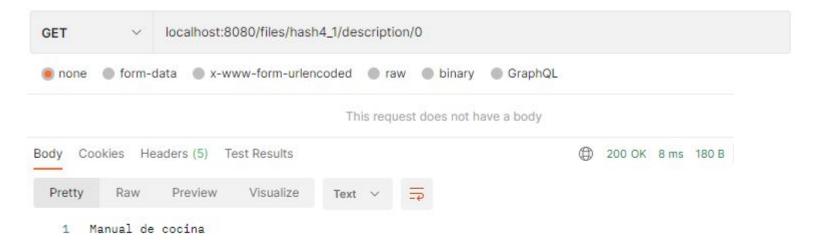
Get File Keyword By Hash



Get File Title By Id



Get File Description By Id



Get File Keyword By Id



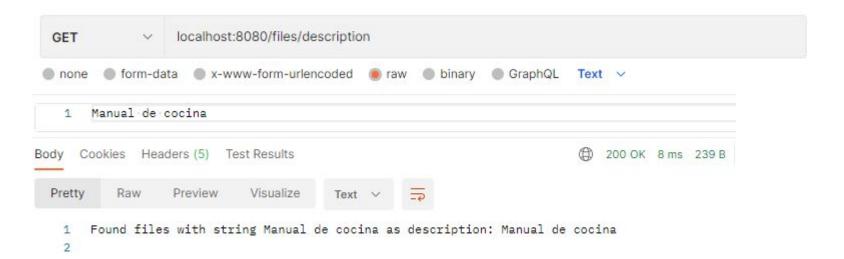
Search By Anytype



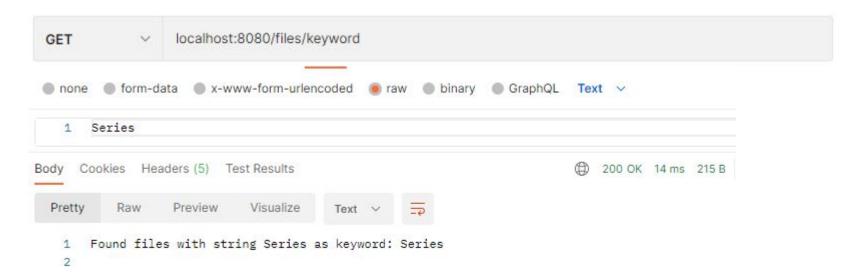
Search By Title



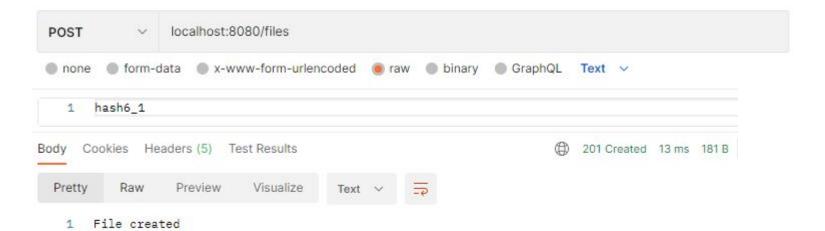
Search By Description



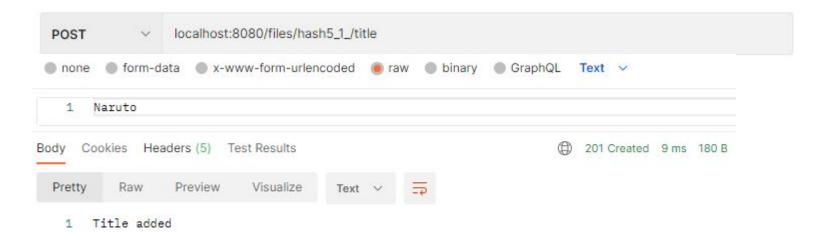
Search By Keyword



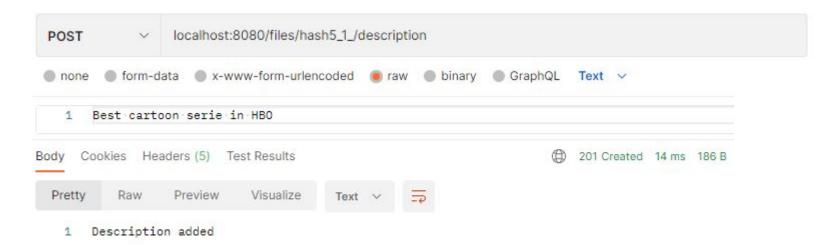
Add File



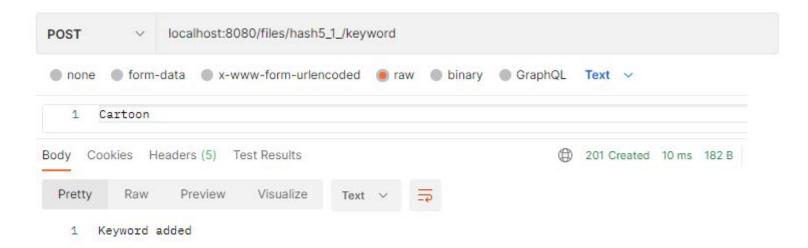
Add File Title



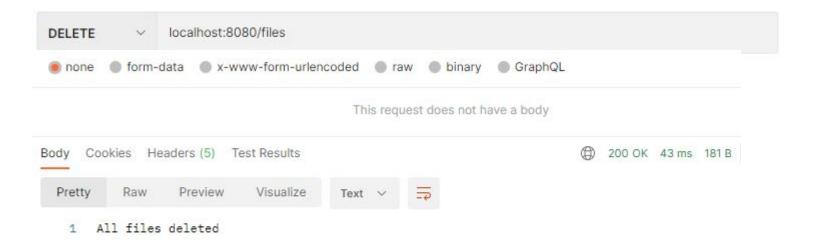
Add File Description



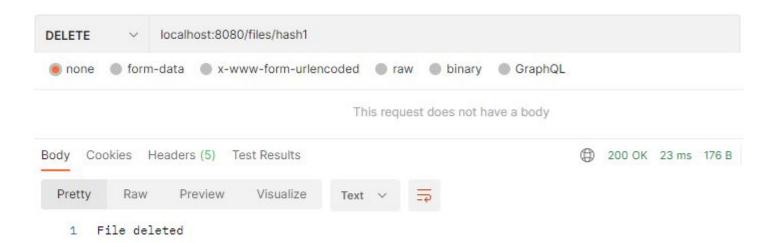
Add File Keyword



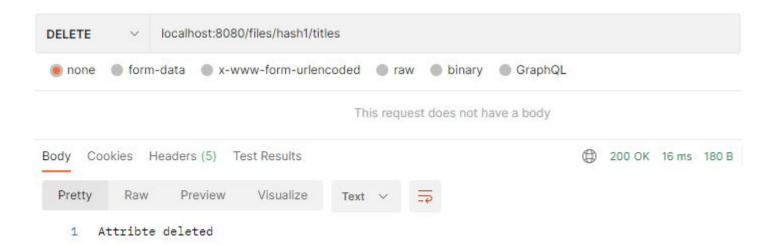
Delete All Files



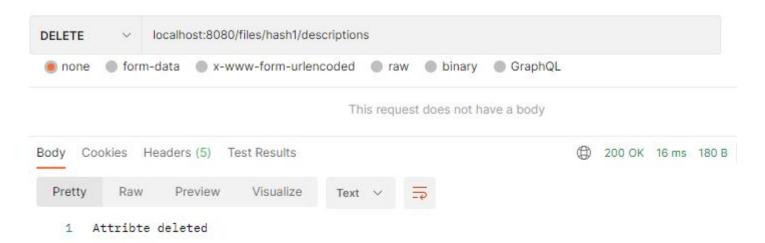
Delete a File



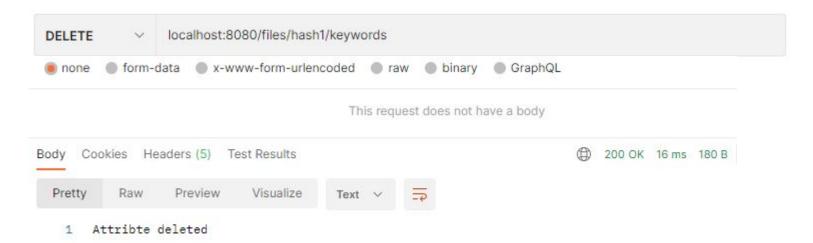
Delete File Titles



Delete File Descriptions



Delete File Keywords



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