

# Project documentation: **Local FileSystem Explorer**



**Authors:**

Robert Mihai  
Guilherme Carra



# Table of content

<b>URL Git repository at code.assembler</b>	<b>3</b>
<b>Requirements</b>	<b>4</b>
<b>Instructions</b>	<b>6</b>
<b>Incidents Record</b>	<b>7</b>

URL Git repository at code.assembler

- <https://code.assemblerschool.com/guillherme-carra/Local-FileSystem-Explorer>

## Requirements

- **Create, modify and delete directories**
- **Browse through directories** from an **initial path**
  - The **initial path** will start from a **folder** inside the **project repository** whose name will be "**root**".
- **Search directories and files by name**
  - In the case of **searching for files by name**, you must also be able to specify **their extension as part of the name**.
- **Navigate through the initial path** established and all the folders created from that path. Therefore the **user will not be able** to see or **navigate** to the **parent folders of the "root" folder**.
- **Upload a file** to a directory
- **See** the following **information** of **files** and **directories**
  - Creation date
  - Last Modified Date
  - Extension (if it's a file)
  - Size
    - If it is less than **1 MB** show **KB**, otherwise show **MB**
- **Show** the **icon** of the main **file extensions** such as:
  - doc
  - csv
  - jpg
  - png
  - txt
  - ppt

- odt
  - pdf
  - zip
  - rar
  - exe
  - svg
  - mp3
  - mp4
- **View** the uploaded **images**.
- **Play** the uploaded **videos**.
- **Play** the uploaded **audios**.

## Instructions

- Double click to **enter** a folder on the folder's content screen.
- Right click to **cut** a file/folder and then right click again to paste it somewhere.
- Right click to **rename** a file or folder.
- Right click and **delete** to send a file/folder to the **Trash**.
- Right click and **delete** on a file/folder that is on Trash to **permanently** exclude it.
- Type something on the **search** bar and press **enter** to search a file/folder.
- Click on the top right icon to **upload** a file to the **current** opened folder

## Incidents Record

#1	Issues with initial pull/push of commits in Git
<b>Description of the incident:</b>  Problems with the creation of branches and pulling/pushing commits at the beginning of the project.	
<b>How was the issue resolved?</b>  Doing some tests and creating the branch structure all together.	
<b>What lessons could we learn from the incident?</b>  We still need more practice to gain confidence with Git.	

#2	Work with one php file for all functions
<b>Description of the incident:</b>  Problems with how to implement all the functions using only one PHP file.	
<b>How was the issue resolved?</b>  Use data parameters on ajax requests to send a \$_POST variable specifying what operation must be executed and filter it using a switch statement on PHP.	
<b>What lessons could we learn from the incident?</b>  Use ajax data to provide information to PHP.	



<b>#3</b>	Undesired return values with scandir()
<b>Description of the incident:</b>	
While using scandir() it normally returns folders with "." or ".." names.	
<b>How was the issue resolved?</b>	
It was used the array_diff() to remove these values.	
<b>What lessons could we learn from the incident?</b>	
How array_diff() and scandir() work.	

<b>#4</b>	How to return a processed CSV file from PHP to Javascript
<b>Description of the incident:</b>	
It is simple to read a CSV file with PHP but it was confusing to send this processed CSV to Javascript.	
<b>How was the issue resolved?</b>	
The solution found was to make a string containing a HTML table inserting each CSV line on this string variable and at the end return this string with echo.	
<b>What lessons could we learn from the incident?</b>	
It is possible to create a whole HTML element with PHP and append it with Javascript later.	

#5	How to implement the search bar for a recursive search
<b>Description of the incident:</b>  It was known that scandir() returns all the elements in a folder, but it does not work to search for a specific file in all the folders inside the root.	
<b>How was the issue resolved?</b>  It was found the PHP class DirectoryIterator which does this task returning all folders and files inside the target folder.	
<b>What lessons could we learn from the incident?</b>  How to use DirectoryIterator and RecursiveIteratorIterator.	

#5	How to upload a file and specify the target folder to upload it
<b>Description of the incident:</b>  When uploading a file you can not send anymore data with it.	
<b>How was the issue resolved?</b>  It was inserted on the ajax URL the target folder as a GET parameter so PHP could use it to upload the file to the right folder	
<b>What lessons could we learn from the incident?</b>  File upload request does not accept anymore data on the ajax statement.	