

Local FileSystem Explorer

Official Documentation

Authors:

Albert Grandes

Eloy Rodriguez

1. Introduction	3
1.1. Project Idea	3
1.2. Main objectives	3
2. Folder organization	4
3. Flow of program	5
4. Issues in development	6
2.1. FrontEnd	6
2.2. BackEnd	6
2.3. Others	6
5. Wireframe history	7
5.1 Original wireframe	7
5.2 First version (rejected)	8
5.3 Dark version	8
6. Other tools	10
6.1 NPM	10
6.2 JQuery	10

1. Introduction

1.1. Project Idea

This project main objective is create a system file explorer that allows the user to navigate, create directories and upload files in the same way as he would in his usual operating system.

The file explorer is a tool that allows you to directly view and manipulate the files and directories associated with a path, so you must take into account from which path the user starts and which path they can access.

In our team we try to do a minimalist and functional file system based in a simple division in frontend and backend that one (backend) is the responsible to send the file and folder data in JSON format and the other (frontend) is in charge to transform this information in something visual for the user.

1.2. Main objectives

1. Understand how the file system works.
2. Improve your knowledge in PHP (special the use of files inside a server)
3. Improve your knowledge in HTML, CSS & Javascript (the presentation and use of data provided by the server)
4. Improve your knowledge in logic and programming.

2. Folder organization

Local-FileSystem-explorer - The root of the project

- **html** - The folder that contains the php of basic elements in html to build index.php
 - *Footer.php*
 - *Head.php*
 - *header.php*
- **images** - This folder have all the images used in the folder show container.
- **node_modules** - This is a folder of NPM. Inside we have JQuery and Bootstrap dependencies. [Inside .gitignore]
- **php** - This folder contains all the php of the page that make orders in the server side.
 - *csvReaderToHTML.php*
 - *deleteFile.php*
 - *dirStruct.php*
 - *fileInfo.php*
 - *newFolder.php*
 - *openFile.php*
 - *renameFile.php*
 - *rootStruct.php*
 - *searchEngine.php*
 - *uploadFile.php*
- **root** - This folder contains all the information that show to the user.
- **src** - This folder contains the personal _js- and css of the page.
 - *style.css*
 - *app.js*
- **.gitignore**
- *index.php*
- **README.md**
- **package.json**
- **package-lock.json**

3. Flow of program

The program have a very simple flow for the user and internal. The program is very intuitive you can navigate easy making click to the folders and can try to look or download a file making click over there.

Internally the program function in the frontend with calls to backend. The backend make the response in json that is the devolution that use the client side to build the content in HTML.

This allow a very big division between backend and frontend that make easy the evolution separately of the program.

4. Issues in development

4.1. FrontEnd

- Problems with Bootstrap to make the web page design responsive. We ruled out the use of Bootstrap for problems in creating the desired style.
- We overused javascript in which all the presentation of the files fell in the user's view.
- We studied the use of tree view for our tree folder in the right of our webpage.
- We try to use media queries to make responsive the web page.

4.2. BackEnd

- Very simple code, but complex to understand quickly the first time. We need to change some variables to make the code more easy to read.
- We return the json of file information only in string. We change this behavior with the use of **header('Content-type: application/json')**.
- We have problems creating functions with existing name in PHP. It is important to know a little more or investigate if a function name is reserved or not by php.

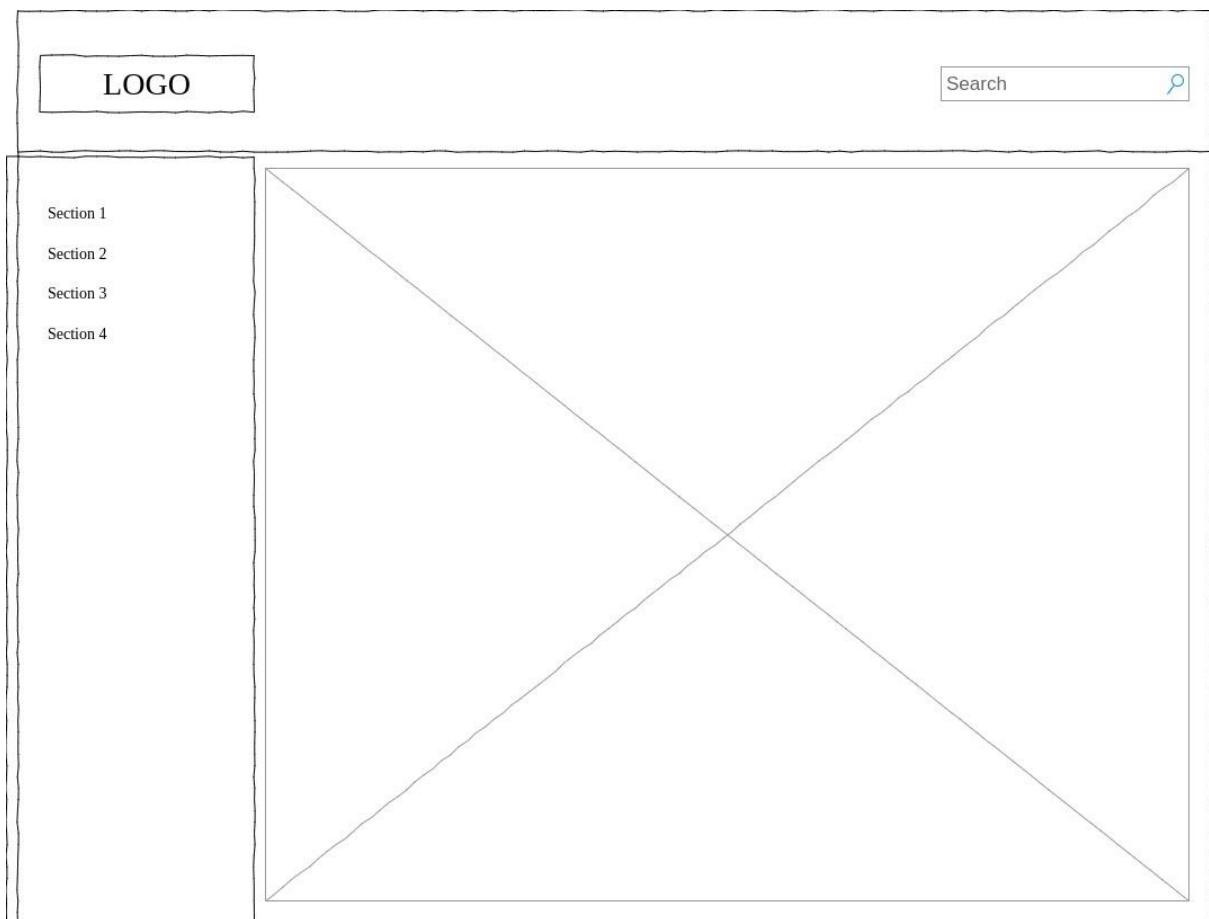
5. Wireframe history

Now we can look a little history of our pages design development.

5.1 Original wireframe

We try to create a very minimalistic web page that have the root folder tree in the left and the content of the folder in the right. A very simple logo in the header left and a find search bar in the header right.

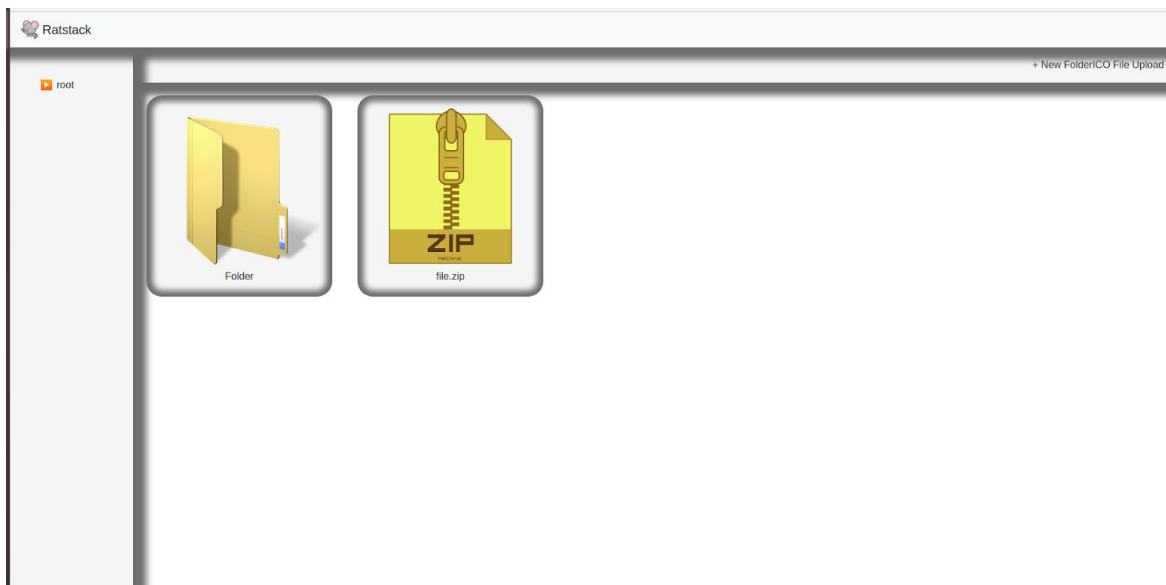
A simple wireframe to make easy to add some new characteristics that can appear during the project evolution.



5.2 First version (rejected)

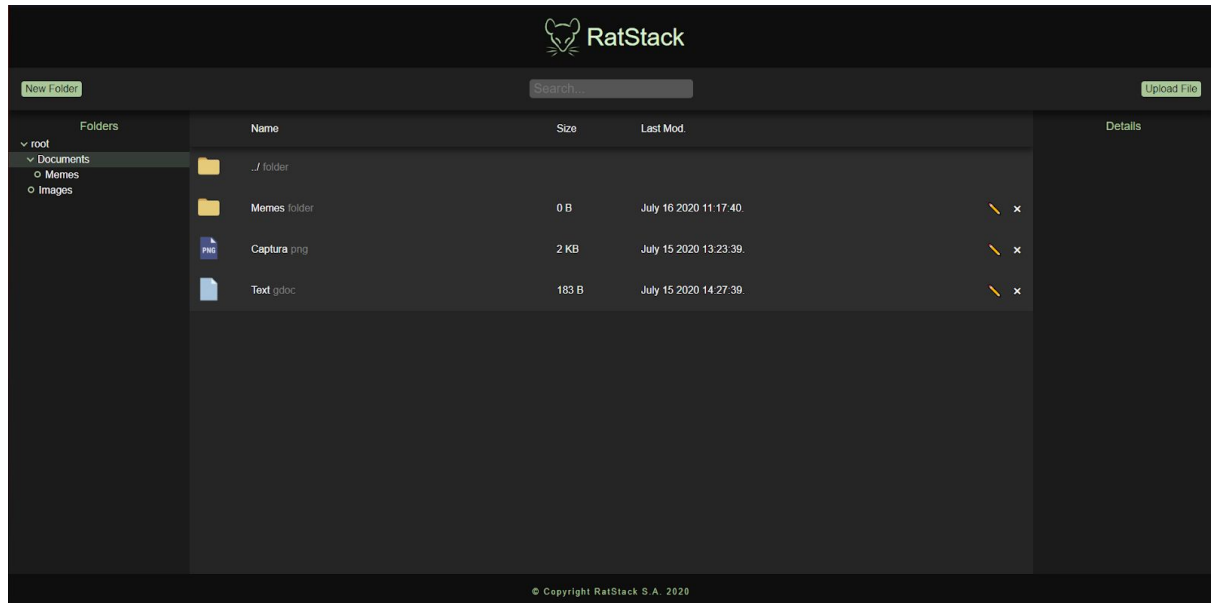
In this versión we try to make a minimalistic and with shadow web page. We are trying to make the sensation of depth like a hole (a rat hole, to make sense the name of webpage Ratstack).

This version was discarded in first reason because was based in a Bootstrap that make problems to make responsive and to do basic changes with our own stylesheet. The second reason was artistic motivations. We found a little old style of webpage and we try to make more modern in the new versión.



5.3 Dark version

In this version we replace the colors and change the position of some elements of webpage. We make a distance for the original wireframe and improvise a little to make a more “young” web page.



6. Requirements check

- You will not be able to use global variables in PHP.
- You must use GIT
- You must use the PHP v7
- Create a clear and orderly directory structure
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use inline styles
- In the case of using different programming languages always define the implementation in separate terms
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project

7. Other tools

In this project we use other tools and libraries for a better result of the project.

7.1 NPM

Npm is the world's largest Software Registry, the registry contains over 800,000 code packages. Open-source developers use npm to share software. Many organizations also use npm to manage private development.

In this project we use NPM to call Bootstrap and JQuery in our project. The importations are in/node_modules/ and you can see in the git repository. In the final version we discard bootstrap in our project.

7.2 JQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

We use jQuery to make easier ajax request in the webpage. At the same time, we use to some other design functionalities.