Redirecting your old website to a new one

Redirecting your website can be useful in several scenarios. For example, if for some reason your domain name has changed from <https://example.com> to <https://new_example.com> you would most likely want to redirect users that try to search for the old URL to the new one.

Another scenario that I had to deal with is changing your directory structure. When I first started my website I had all my files in one folder, but after some time I decided to change the location of the files and organize them better. For example, the HTML file for one of the articles I have posted on my website was present in the root directory as shown in the image below:

Graphical user interface, text, application

Description automatically generated

Opening that page would correspond to this URL: <https://bojanstavrikj.github.io/wunderground_scraper>. At some point my root directory started getting quite messy, so I decided to organize the content part of my page into different folders as shown below.

Graphical user interface, text, application

Description automatically generated

Once this move was implemented, the URL for the same page changed to: <https://bojanstavrikj.github.io/content/page1/wunderground_scraper>. Therefore, trying to open the old link in a web browser would throw an error.

Assuming, the old link has been optimized for search engine optimization (SEO) and is already bringing in traffic to your website you would probably like to redirect any user trying to access the old URL to the new one. There are several ways of doing this.

Keep in mind that the original files still need to be online at the original location. So using my case as an example, if you look at the second image above my wunderground\_scraper.html page file is present both in the root directory and in the new directory. Therefore, the changes implemented for points 1 and 2 below will be done on the file in the root directory.

1. Redirect old to new URL using HTML

One of the simplest ways of implementing this is by instructing the web browser to redirect users to a new page using a meta tag in the header of your HTML page. For my case, I would open the wunderground\_scraper.html file from my root directory (first image) and use the following code:

<meta http-equiv="refresh" content="0";url='https://bojanstavrikj.github.io/content/page1/wunderground\_scraper'">.

This is a pretty easy way of solving the redirect. Although, as WS3 point out some web browsers get confused so it’s not recommended.

1. Redirect old to new URL using Javascript

This is another easy and better way of doing the redirect. Going back to my example, I used the following code in the header of my page:

<script>  
// Simulate a mouse click:  
window.location.href = "https://bojanstavrikj.github.io/content/page1/wunderground\_scraper";  
  
// Simulate an HTTP redirect:  
window.location.replace("https://bojanstavrikj.github.io/content/page1/wunderground\_scraper");  
</script>

This would also achieve the same result. And now if you search in your browser for this page: <https://bojanstavrikj.github.io/wunderground_scraper> you would be redirected to <https://bojanstavrikj.github.io/content/page1/wunderground_scraper>.

1. Redirect old to new URL using .htaccess file

The last option I will show in this article is to use a .htaccess file in order to tell the browser to redirect your site. You first need to (1) create a .htaccess file in your root directory. In this file all you need to write is the following lines:

In my case the old path was /wunderground\_scraper, and the new path is /content/page1/wunderground\_scraper. Therefore, for this example the redirect would be:

Redirect /wunderground\_scraper <https://bojanstavrikj.github.io/content/page1/wunderground_scraper>

Note that you need to leave a space between the Redirect, old and new path to the website.