



ONLINE
ADA

+1 (541)-999-1687
207 E 5th Ave Ste 240
Eugene, OR, 97401

WordPress Remediation With a Custom Plugin

A practical guide to remediating WordPress websites

Overview

WordPress is an open source content management platform. Not surprisingly, it's also the most-used content management platform in the history of the Internet. This means that as an agency, there's a good chance that you're managing or coming across WordPress websites.

Being open source, WordPress allows you virtually limitless possibilities. There's a massive developer community that builds and maintains thousands of plugins for the platform.

For digital accessibility remediation, however, the open nature of WordPress will present some problems.

- Developers often customize 3rd-party plugins, only to find their custom code overwritten when the plugin developer pushes a version release.
- WordPress itself is almost always customized, but platform updates to WordPress can overwrite that customization.

We're seeing many WordPress website owners frustrated with this state of affairs, because they've paid their developer to remediate a website, only to see their money wasted when a WordPress update essentially deletes any remediation that has been done to the website.

The solution: build a plugin to store your remediated code.

Our development team has devised and successfully tested a novel idea that allows remediated changes to stay in place, uninterrupted, even when WordPress and its plugins are updated over time.

- This will save the website owner thousands of dollars annually. -

The concept is to place the remediated code changes into a plugin. This protects the remediation from WordPress updates and it makes any future remediation very easy to manage and maintain.



Step-By-Step Instructions

Step 1)

Create a folder and name it using whatever naming conventions you are using now. We use the format "onlineada-[WEBSITE NAME]-remediation". For example, for ESPN.com it would be "onlineada-espn-remediation". The name of this folder is important only for identifying it by your staff in the future.

Step 2)

Inside this newly created folder create another folder called "assets", a file called "index.php", and a file called "style.css".

- **Assets folder:** stores images, page-specific CSS files, and JavaScript scripts
- **index.php:** This is the registration file for the plugin. At the top is the definition so WordPress can read the plugin followed by an initialization PHP code necessary to make the rest of the plugin work. Our plugin is very simple, so the only thing we will add to this file is a WordPress hook to enqueue our JavaScript and CSS, which we will describe below.
- **style.css:** This is a global CSS file allowing for the application of styles across all pages. In truth, it is no different than the CSS files in the assets / CSS folder and must be registered the same way, but structurally we use this file globally so we know right where to look for any styles being applied to all pages. This is where we would apply a new focus indicator style, for instance, and any supplemental styles that would ensure the new focus indicator looks good. We also use this file for correcting contrast issues in the header and footer.

Step 3)

Inside the "index.php" file, at the very top of the file, write the following:

```
<?php
/*
    Plugin Name: OnlineADA Remediation Plugin
    Plugin URL: https://onlineada.com
    Description: Site-specific remediation fixes for [Website Or Client Name]
    Version: 1.0
    Author: OnlineADA
    License: GPL2
*/
```



Make sure to include that opening "<?php" code. It's very important. The rest is a block comment in PHP for defining specific bits of information that will be read by WordPress to display to a user on the Plugins page of the WordPress Admin Dashboard.

After the above lines, add the following code:

```
add_action("wp_enqueue_scripts", function(){
    $assets = plugins_url("/assets", __FILE__);
    wp_enqueue_style("global-styles", plugins_url("/style.css", __FILE__));
    wp_enqueue_script("oada-global", $assets . "/js/global.js", ["jquery"], false, true);

    $js_files = scandir(__DIR__ . "/assets/js");

    if( is_search() ){
        $path = $assets . "/js/search-results.js";
        wp_enqueue_script("oada-search", $path, ["jquery"], false, true);
        $path = $assets . "/css/search-results.css";
        wp_enqueue_style("search-styles", $path);
    }

    $skip = [
        ".",
        "..",
        "global.js",
        "search-results.js",
    ];

    foreach($js_files AS $file){
        if(in_array($file, $skip)){
            continue;
        }

        $slug = explode(".", $file)[0];
        $path = $assets . "/js/{$file}";

        if(is_page($slug)){
            wp_register_script("oada-{$slug}", $path, ["jquery"], false, true);
            wp_localize_script("oada-{$slug}", "remediation_vars", [ "assets" => $assets ]);
            wp_enqueue_script("oada-{$slug}");
        }
    }
}
```



```
$css_files = scandir(__DIR__ . "/assets/css");
$skip = [
    ".",
    "..",
    "search-results.css",
];

foreach($css_files AS $file){
    if(in_array($file, $skip)){
        continue;
    }
    $slug = explode(".", $file)[0];
    $path = $assets . "/css/{$file}";
    if(is_page($slug)){
        wp_enqueue_style("oada-{$slug}", $path, false);
    }
}
}, 99);
```

Step 4)

Inside the "assets" folder, add the following folders: "css" , "img" , "js".

Step 5)

Inside the CSS folder you can now add as many files as you like. Each file represents one page of the site. These files **MUST** be named the same as the slug of the page.

For example: if a page exists at the URL "https://example.com/contact-us" , the important part of the URL is the end part "contact-us". This is the slug of the page, which can also be found in the admin dashboard on the edit screen for each page under the right-hand section "permalink".

The reason this is important is because the file **NEEDS** to be the same name as the slug. Going back to the example above where the slug of the page is contact-us, the name of the CSS file needs to be "contact-us.css".



Step 6)

Repeat the exact same instructions in Step 5, but this time for the JavaScript files instead of the CSS files. These will live inside the JS folder. The only difference here is that you also want to create one file called "global.js" that, just like style.css, will be loaded on each and every page.

Step 7)

The last step is to zip up your top level folder, the first one you created called "onlineada-[WEBSITE NAME]-remediation" using WinRAR or 7zip or some other zip software, and then in the admin dashboard of the website, go to the plugins section and upload your plugin. Lastly, activate it.