

```

1  <?php
2
3
4      /*
5      *
6      *   First, check to see if the form has been filled out.
7      *   If not (!isset), do nothing
8      *
9      */
10
11     if (!isset($_POST['token'])) {
12         $atRun = "<h3>Canvas De-Clutter Tool</h3>";
13     } else {
14
15     /*
16     *   If there is some form data, begin processing it.
17     */
18
19     echo "<p>Canvas limits the number of things you can do per minute
20         with a remote tool like this. To avoid overwhelming that limit,
21         there's a 1-2 second pause for each file, assignment, page or link
22         that's affected. This page is the <em>only</em> record that will
23         exist of your De-Cluttering. When you close this window/tab, it
24         will be gone. Please copy/paste the records below if you think you
25         will need the assignment names or any other info.</p>
26         <ol>";
27
28     /*
29     *   These next few lines just deal with the URL and the authorization
30     *   token you provided. courseNum figures out your course's unique ID
31     *   from your URL (it's the number at the end).
32     */
33
34     $courseNum = explode("/", $_POST['url'])[4];
35     $token = $_POST['token'];
36
37     $authorization = "Authorization: Bearer " . $token;
38     $url = "https://oit.instructure.com/api/v1/courses/$courseNum";
39
40     $headers = array(
41         $authorization);
42
43     /*
44     *   This next bit finds and deletes all of your course's Pages.
45     *   The sleep() command you'll see throughout this code is what
46     *   makes this take some time. We need to limit the number of access
47     *   requests we make, or else Canvas will freeze you out for a little while.
48     */
49
50     // Get all of the course's pages
51     sleep (1);
52     $curl = curl_init(); // start cURL object to get a list of pages
53
54     curl_setopt($curl, CURLOPT_URL, $url . "/pages"); // Get all course
55     pages
56     curl_setopt($curl, CURLOPT_HTTPGET, 1); // Use the GET method
57     curl_setopt($curl, CURLOPT_HTTPHEADER, $headers); // Give Canvas
58     your access token
59     curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE); // Don't print a
60     bunch of stuff on the screen
61
62     $result = curl_exec($curl);
63
64     $pages = json_decode($result, true);
65     curl_close($curl);
66
67
68

```

```

59     $targets = array();
60
61     foreach ($pages as $page) {
62         $t1 = explode("/", $page['html_url']);
63         $t = $t1[count($t1) -1];
64         array_push($targets, $t);
65     }
66
67     /*
68     * This is where deletion begins. The foreach loop gets every
69     * page URL from our last request and deletes each page individually.
70     */
71
72     $pgCount = 0;
73     foreach ($targets as $page) {
74         sleep(1);
75         $curl = curl_init(); // Start cURL object to delete pages
76
77         echo "<li>Deleting $page . . . </li>";
78
79         curl_setopt($curl, CURLOPT_URL, $url . "/pages/" . $page);
80         curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "DELETE"); // Using
DELETE method
81         curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
82
83         $result = curl_exec($curl);
84
85         curl_close($curl);
86         $pgCount++;
87     }
88
89     // Pages have all been deleted
90
91     /*
92     * Next, blank out all assignment descriptions and replace the
93     * assignment name with a number. The process is pretty similar.
94     * If a CSET person is reading this, please don't get angry that I
95     * didn't decompose this better.
96     */
97
98
99     // Get course assignments
100    sleep (1);
101    $curl = curl_init(); // start cURL object to get a list of pages
102
103    curl_setopt($curl, CURLOPT_URL, $url . "/assignments"); // Get all
course pages
104    curl_setopt($curl, CURLOPT_HTTPGET, 1);
105    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
106    curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
107
108    $result = curl_exec($curl);
109
110
111    $assignments = json_decode($result, true);
112    curl_close($curl);
113
114    /*
115    * The section below is where name and description modification
116    * happens. We are not deleting assignments because that would delete
117    * student submissions and existing grades.
118    */
119
120    $assgCount = 1; // For assignment names
121    foreach ($assignments as $assignment) {
122        sleep(1); // Pause execution for 1 second on each page to avoid
overwhelming the API request limit

```

```

123
124     echo "<li>Changing " . $assignment['name'] . " to Assignment
$assgCount . . . </li>";
125
126     $body = array(
127         'assignment[name]' => "Assignment " . $assgCount,
128         'assignment[description]' => "<p>&nbsp;</p>",
129         'assignment[published]' => FALSE // or TRUE
130     );
131     $assgCount++; // Increment for next assignment
132
133     $curl = curl_init(); // Start cURL object to change assignments
134
135     curl_setopt($curl, CURLOPT_URL, $url . "/assignments/" .
$assignment['id']);
136     curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "PUT"); // Using PUT
method
137     curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
138     curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($body));
//Arrange post data
139     curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
140
141
142     $result = curl_exec($curl);
143
144     curl_close($curl); //
145 }
146
147
148 /*
149 * Up next, we'll delete files. This works very similar to
150 * the page deletion step.
151 */
152
153
154
155     sleep (1);
156     $curl = curl_init(); // start cURL object to get a list of pages
157
158     curl_setopt($curl, CURLOPT_URL, $url . "/files"); // Get all course
pages
159     curl_setopt($curl, CURLOPT_HTTPGET, 1);
160     curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
161     curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
162
163     $result = curl_exec($curl);
164
165
166     $files = json_decode($result, true);
167     curl_close($curl);
168
169     $fileCount = 0;
170     foreach ($files as $file) {
171         sleep(1);
172         $curl = curl_init(); // Start cURL object to delete pages
173
174         $body = array(
175             'file[replace]' => TRUE
176         );
177
178         echo "<li>Deleting " . $file['filename'] . " . . . </li>";
179
180         curl_setopt($curl, CURLOPT_URL,
"https://oit.instructure.com/api/v1/files/" . $file['id']);
181         curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "DELETE"); // Using
DELETE method
182         curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);

```

```

183     curl_setopt($curl, CURLOPT_POSTFIELDS, $body);
184     curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
185
186     $result = curl_exec($curl);
187
188
189
190     curl_close($curl);
191     $fileCount++;
192 }
193
194
195
196 // The quiz logic has too many steps to complete as of 4/22. If
// there's enough demand, I'll do it before Monday.
197
198     $_POST = null;
199     $curl = null; //Empty out all variables, just in case
200     $pages = null;
201     $assignments = null;
202     $token = null;
203     $url = null;
204
205 }
206
207 /*
208 * This last piece is just a little message to let you know how much
209 * stuff you de-cluttered!
210 */
211
212     $atRun = "<p>I deleted " . $pgCount . " of the Pages at that Canvas
course. " . $assgCount . " assignments have          been renamed
213         'Assignment #' and given blank descriptions. " .
        $fileCount . " files were deleted.</p>";
214
215     ?>
216
217
218 <!-- This is where the webpage content begins -->
219     <head>
220         <script>
221             function turnTheKeys(){
222                 var chk = document.getElementById("confirm").checked;
223                 var sb = document.getElementById("btnSubmit");
224                 sb.disabled = !chk;
225             }
226         </script>
227     </head>
228
229     <form action="" method="post" enctype="application/x-www-form-urlencoded">
230         <?php
231             echo $atRun;
232             ?>
233
234         <p>
235             Please provide the URL of your Canvas shell's home page: <input type
="text" name="url" placeholder=
                "https://oit.instructure.com/courses/####">
                </p>
236
237
238         <p>
239             Please provide your Canvas Authorization Token: <input type="text"
name="token" placeholder="It's a lot of text and symbols">
                </p>
240
241
242         <p>
243             <input type="checkbox" id="confirm" onChange="turnTheKeys()"><b>Yes

```

```
244         </b>, I have already backed up all of the course content I want. I
           understand
245         that this deletion is <b>irreversible</b>, and that any content
           I want to restore will have to come from some backup source.
246     </p>
247     <input id="btnSubmit" type="submit" value="De-Clutter" disabled=
           "true">
248 </p>
249 <p>
250     Please note that Canvas limits the rate at which you can perform
           requests like this. This tool will process one page or assignment
           every second.
251 </p>
252 </form>
253
```