



index : administration/savane.git

frontend ▾ switch

Savannah's Savane

summary refs log tree commit diff

log msg ▾ search

path: root/frontend/php/file.php

blob: d88c0b94e31b5b4083493d36665b6967b43b0d7c (plain)

```
1 <?php
2 # Provide an URL with a valid filename that browsers will use (save as...)
3 #
4 # Copyright (C) 1999, 2000 The SourceForge Crew
5 # Copyright (C) 2001, 2002 Laurent Julliard, CodeX Team, Xerox
6 # Copyright (C) 2000-2006 Mathieu Roy <yeupou--gnu.org>
7 # Copyright (C) 2014, 2016, 2017 Assaf Gordon
8 # Copyright (C) 2001-2011, 2013, 2017 Sylvain Beucler
9 # Copyright (C) 2013, 2014, 2017-2026 Ineiev
10 #
11 # This file is part of Savane.
12 #
13 # Code written before 2008-03-30 (commit 8b757b2565ff) is distributed
14 # under the terms of the GNU General Public license version 3 or (at your
15 # option) any later version; further contributions are covered by
16 # the GNU Affero General Public license version 3 or (at your option)
17 # any later version. The license notices for the AGPL and the GPL follow.
18 #
19 # Savane is free software: you can redistribute it and/or modify
20 # it under the terms of the GNU Affero General Public License as
21 # published by the Free Software Foundation, either version 3 of the
22 # License, or (at your option) any later version.
23 #
24 # Savane is distributed in the hope that it will be useful,
25 # but WITHOUT ANY WARRANTY; without even the implied warranty of
26 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
27 # GNU Affero General Public License for more details.
28 #
29 # You should have received a copy of the GNU Affero General Public License
30 # along with this program. If not, see <https://www.gnu.org/licenses/>.
31 #
32 # Savane is free software: you can redistribute it and/or modify
33 # it under the terms of the GNU General Public License as
34 # published by the Free Software Foundation, either version 3 of the
35 # License, or (at your option) any later version.
36 #
37 # Savane is distributed in the hope that it will be useful,
38 # but WITHOUT ANY WARRANTY; without even the implied warranty of
39 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
40 # GNU General Public License for more details.
41 #
42 # You should have received a copy of the GNU General Public License
43 # along with this program. If not, see <https://www.gnu.org/licenses/>.
44
45 $GLOBALS['skip_csp_headers'] = 1;
46
47 foreach (['init', 'http', 'form-check'] as $i)
48     require_once ("include/$i.php");
49
50 function file_exit ($func, $param)
51 {
52     unset ($GLOBALS['skip_csp_headers']);
```

```
53     utils_set_csp_headers ();
54     $func = "exit_$func";
55     $func ($param);
56 }
57
58 function file_exit_error ($str)
59 {
60     file_exit ('error', $str);
61 }
62
63 extract (sane_import ('request',
64     [
65         'preg' => [['file_id', '/^\(\\d+|test[.]png)$/' ]], 'digits' => 'file_uid'
66     ]
67 ));
68
69 if (!$file_id)
70     file_exit ('missing_param', ['file_id']);
71
72 if ($file_id == 'test.png')
73     {
74         header ('Content-Type: image/png');
75         $fname = $GLOBALS['sys_www_topdir'] . '/images/common/floating.png';
76         header ('Content-Length: ' . stat ($fname)['size']);
77         header ("Content-Disposition: attachment; filename=$file_id");
78         readfile ($fname);
79         exit (0);
80     }
81
82 # Check privacy of the item this file is attached to and reject access by
83 # non-authorized users.
84
85 $result = db_execute (
86     "SELECT item_id, artifact FROM trackers_file WHERE file_id = ?", [$file_id]
87 );
88
89 if (db_numrows ($result) > 0)
90     {
91         $item_id = db_result ($result, 0, 'item_id');
92         $artifact = db_result ($result, 0, 'artifact');
93     }
94 else
95     # TRANSLATORS: the argument is file id (a number).
96     file_exit_error (sprintf (_("File #%s not found"), $file_id));
97
98 $in = [0 => $artifact];
99 $out = [];
100
101 if ($sane_sanitizers['artifact'] ($in, $out, 0, null))
102     {
103         # TRANSLATORS: the argument is artifact name ('bugs', 'task' etc.)
104         $str = sprintf (_('Invalid artifact %s'), "<em>$artifact</em>");
105         unset ($artifact);
106         file_exit_error ($str);
107     }
108
109 function assert_file_access ($item_fields, $file_uid)
110 {
111     if ($item_fields['privacy'] != '2')
112         return;
113     if (user_can_be_super_user ($file_uid))
114         # We are in the file domain and have no access to cookies, so we can't tell
115         # if the user has become a superuser; therefore, we let site admins access
116         # any files in any case.
117         return;
```

```

118 $group_id = $item_fields['group_id'];
119 if (!member_check_private ($file_uid, $group_id))
120     file_exit_error (
121         _("Non-authorized access to file attached to private item")
122     );
123 form_check_id ();
124 }
125
126 $item_fields = utils_find_item (
127     $artifact, $item_id, ['privacy'], 'file_exit_error'
128 );
129 assert_file_access ($item_fields, $file_uid);
130
131 $result = db_execute ("
132     SELECT description, filename, filesize, filetype, date
133     FROM trackers_file WHERE file_id = ? LIMIT 1", [$file_id]
134 );
135
136 if (!db_numrows ($result))
137     file_exit_error (
138         sprintf (_("Couldn't find attached file #%.s."), $file_id)
139     );
140
141 $row = db_fetch_array ($result);
142
143 if ($row['filesize'] < 0)
144     file_exit_error (
145         sprintf (_("Attached file #%.s was lost."), $file_id) . " "
146             . sprintf (
147                 _("File attributes: name '%s', size %, type '%s', date %s."),
148                 $row['filename'], $row['filesize'], $row['filetype'],
149                 utils_format_date ($row['date'])
150             )
151     );
152 $mtime = $row['date'];
153 http_exit_if_not_modified ($mtime);
154 header ('Last-Modified: ' . date ('r', $mtime));
155
156 # Check if the filename in database matches the one in the URL.
157 # We do not want to allow broken URL that may make a user download
158 # a file with a given name like "myimage.png" when actually downloading
159 # something completely different like "mystupidvirus.scr".
160 if ($row['filename'] != basename (rawurldecode ($_SERVER['PHP_SELF'])))
161     file_exit_error (
162         _("The filename in the URL does not match the filename "
163             . "registered in the database")
164     );
165
166 $path = "$sys_trackers_attachments_dir/$file_id";
167 if (!is_readable ($path))
168     file_exit_error (_("No access to the file.));
169
170 $row['agpl'] = trim (git_agpl_notice ('This file is served with Savane.));
171
172 # Serve the file with respective attributes.
173 $headers = [
174     'filetype' => 'Content-Type: ', 'filesize' => 'Content-Length: ',
175     'filename' => 'Content-Disposition: attachment; filename=',
176     'description' => 'Content-Description: ', 'agpl' => 'Source-Code-Offer: '
177 ];
178 foreach ($headers as $field => $h)
179     {
180         if (empty ($row[$field]))
181             continue;
182         $val = str_replace (["\n", "\r"], ' ', $row[$field]);

```

```
183 |     header ("h$val");  
184 |   }  
185 | readfile ($path);  
186 | exit (0);  
187 | ?>
```

generated by cgit v1.3 (git 2.53.0) at 2026-06-20 21:05:01 +0000