



# **“Evolette”**

## **Premium Template By iDangero.us**

### **Documentation**

Thank you for purchasing this item. If you have any questions that are beyond the scope of this help file, please feel free to contact us via support ticket form [here](#).

# Table of Contents

---

1. About “Evolette”
2. Evolette Files & Folders Structure
3. Installation
4. Login Process & Security
5. “Tips”
6. HTML Structure
7. Inner Linking
8. CSS Files and Structure
9. Typography
10. JavaScripts
11. JavaScript Functions API
12. Popup Window
13. Forms Handling
14. Mailing Engine
15. To Do List
16. PSD Files and Structure
17. Licensing Terms

# 1. About “Evolette”

---

**Evolette** is the premium Ajax based Administration Backend template. It's only template, it has not real CMS functionality. It's intended to use as a CMS interface.

The main advantage of this template is that it uses only one page. All content pages are loaded "on the fly", so your users do not need to wait the reloading of entire site. Template is fully animated without the use of Flash technology.

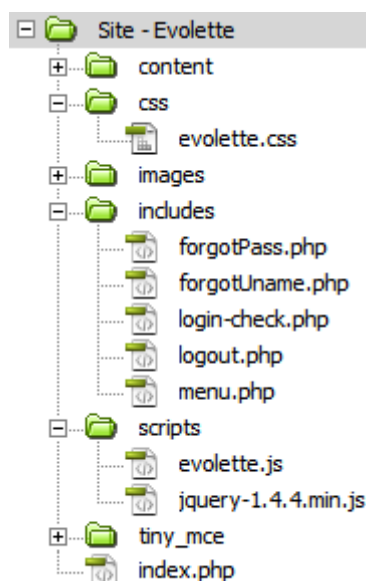
Smooth and good looking animation and Ajax features of this template are realized with the JavaScript jQuery library.

This template is easy to redesign, it has very simple and clear API.

## 2. Evolette Files Structure

---

**Evolette** files are located in the **files/** folder from the downloaded archive:



**content:** in this folder there are .php files with a html content pages.

**css:** folder with a .css files

**images:** here you can find layouts images and icons used in this Template

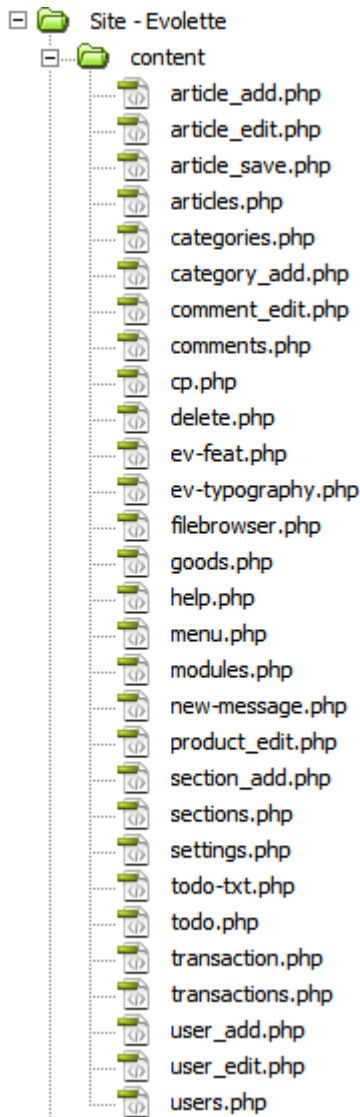
**includes:** 5 php files with menu and 4 scripts used for login and logout.

**scripts:** JavaScripts scripts used in this template

**tiny\_mce:** this folder contains the TinyMCE WYSIWYG text editor

And the main **index.php** in the root folder

Now let's look at the **content/** folder files structure:



As mentioned above this folder contains **php** files with HTML content pages. Every action has its own file: **article\_add.php**, **article\_edit.php**, **article\_save.php** etc

**cp.php** – this file contains HTML code of Control Panel (Home page)

## 3. Installation

---

To install **Evolette** template you have to copy **evolette/** folder to your web-site root folder on the server (It could be a real server or local testing server). And you should to rename it to **administration** (for example).

## 4. Login Process & Security

---

First of all let's look at the principle of login process to know how it works.

When we click on the “Login” button, JavaScript sends **username** and **password** values to the **includes/login-check.php** script file:

```

1 <?php
2 session_start();
3 $username = trim(strip_tags($_POST['username']));
4 $password = trim(strip_tags($_POST['password']));
5 if ($username=="admin" && $password=="demo"){
6     echo "1";
7     $_SESSION['authorized']=true;
8 }
9 else {
10    echo "0";
11    $_SESSION['authorized']=false;
12 }
13 ?>

```

First of all, we have to start session

Then we have to take username and password values posted by JavaScript

After that we need to compare them with correct values. Of course in your real CMS you should make connection with your database and compare with values from this database.

If everything is correct, we have to set "authorized" session variable equal to "true" and to print "1". This "1" will be accepted by JavaScript as a successfully login process, and after that JavaScript will "play" login animation.

Otherwise we have to set "authorized" session variable equal to "false" and to print "0". This "0" will be accepted by JavaScript as an unsuccessfully login process, and after that JavaScript will "show" a message with a wrong login status.

Then in the work process all files with HTML content from the **content/** folder are loaded by JavaScript Ajax requests or included by php scripts. So to protect all our content files we have to add the following code line in their beginning:

```

<?php
session_start();
if(!((($_SERVER['HTTP_X_REQUESTED_WITH'] == 'XMLHttpRequest') || defined('evolette'))
&& $_SESSION['authorized']==true)) exit();
?>

```

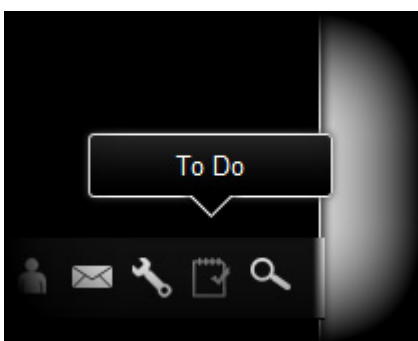
We have to start session

We have to check is the file opened with XMLHttpRequest (Ajax request) or is it included by other php file ("evolette" variable will be defined in the main index.php)

And finally we have to check is the 'authorized' session variable is equal to 'true'

And if all these conditions are NOT satisfied, we have to **exit** from this file

## 5. "Tips"



To use tips for any HTML element you have to add nonstandard attribute TIP or MTIP (for menu links). Here are examples:

Use it in any content pages

```

<a href="index.php" tip="Go to home page">Home</a>

```

```

<a href="index.php" mtip="Go to home page">Home</a>

```

For menu

## 6. HTML Structure

Let's look at the main HTML/PHP structure of Evolette Template. Open index.php file with a text editor (Highly recommended to use Adobe® Dreamweaver® or Notepad ++ editors).

**Index.php** structure:

We have to define "evolette" variable, to protect our content php files from direct access. We have to start session to make available login feature.

Linked CSS and JavaScripts files

Ajax Loader

If "authorized" session' variable is not equal to "true" it means that we are not logged in, so we have to show login page

Logo container

Login form container

Login form container

Wrong login or password message. It is hidden by default

Login form

```
1 <?php
2 define('evolette',true) ;
3 session_start();
4 ?>
5 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
6 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
7 <html xmlns="http://www.w3.org/1999/xhtml">
8 <head>
9 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
10 <link rel="stylesheet" type="text/css" href="css/evolette.css" />
11 <script type="text/javascript" src="scripts/jquery-1.4.4.min.js"></script>
12 <script type="text/javascript" src="tiny_mce/jquery.tinymce.js"></script>
13 <script type="text/javascript" src="scripts/evolette.js"></script>
14 <title>Evolette</title>
15 </head>
16 <body>
17 <div id="ajax-loader">
18 
19 </div>
20 <? if ($_SESSION['authorized']!=true) {
21 <div id="logo">  </div>
22 <div class="lock">
23 <div class="lock2"></div>
24 <div id="logout"></div>
25 </div>
26 <div id="login">
27 <div class="lock-bg"></div>
28 <div class="form-heading">
29 <h3>Administration Login</h3>
30 </div>
31 <div class="login-form">
32 <div class="lf-left">
33 <div class="lf-mid">
34 <div class="wrong-status">Username and password do not match</div>
35 <form name="loginForm" id="loginForm" action="index.php" method="post">
36 <label>Username:<emsp;<emsp;<span class="text-l"><span class="text-r">
37 <input type="text" id="username" size="30" value="admin" />
38 </span></span></label>
39 <label>Password:<emsp;<emsp;<span class="text-l"><span class="text-r">
40 <input type="password" id="password" size="30" value="demo" />
41 </span></span></label>
42 <p>
43 <input type="submit" value="Login" />
44 <span class="forgot"> Forgot your <a href="#" class="openUname">Username</a> or <a href=
45 "#" class="openPass">Password</a> ?</span></p>
46 </form>
47 </div>
48 </div>
```

Container with a Username reminder form.

```
49 <div class="forgotUname">
50 <p>Enter your registered e-mail and we'll send you your username</p>
51 <form method="post" action="javascript:retrieveUsername()" >
52 <span class="text-l"><span class="text-r">
53 <input tip="Only "demo@idangero.us" - will be the right e-mail" type="text" id=
"retrieveUnameEmail" size="20" />
54 </span></span>
55 <input style="margin-left:20px" type="submit" value="Send Username" />
56 </form>
57 <div class="forgotStatus"></div>
58 </div>
59 <div class="forgotPass">
60 <p>Enter your registered e-mail and we'll send you a NEW password</p>
61 <form method="post" action="javascript:retrievePassword()" >
62 <span class="text-l"><span class="text-r">
63 <input tip="Only "demo@idangero.us" - will be the right e-mail" type="text" id=
"retrievePassEmail" size="20" />
64 </span></span>
65 <input style="margin-left:20px" type="submit" value="Send Password" />
66 </form>
67 <div class="forgotStatus">Message with your Username was successfully sent</div>
68 </div>
69 <div class="clear"></div>
70 </div>
71 <div id="maincontent">
72 <div id="menu">
73 <div class="menu"></div>
74 </div>
75 <div class="content">
76 <div class="content-inner"></div>
77 </div>
78 <div class="clear"></div>
79 </div>
80 <?php
81 }
82 if ($_SESSION['authorized']==true) {
83 >
84 <div id="logo" style="left: 240px; margin-left: 0px;">  </div>
85 <div class="lock" style="left: 0px; top: -5px; height: 299px; margin-left: 0px;">
86 <div class="lock2" style="display: none;"></div>
87 <div id="logout" style="display: block;"></div>
88 </div>
89 <div id="login" style="display:none">
90 <div class="lock-bg"></div>
91 <div class="form-heading">
92 <h3>Administration Login</h3>
93 </div>
94 <div class="login-form">
95 <div class="lf-left">
96 <div class="lf-mid">
97 <div class="wrong-status">Username and password do not match</div>
98 <form name="loginForm" id="loginForm" action="index.php" method="post">
99 <label>Username:&nbsp;&nbsp;&nbsp;<span class="text-l"><span class="text-r">
100 <input type="text" id="username" size="30" value="admin" />
101 </span></span></label>
102 <label>Password:&nbsp;&nbsp;&nbsp;<span class="text-l"><span class="text-r">
103 <input type="password" id="password" size="30" value="demo" />
104 </span></span></label>
105 <p>
106 <input type="submit" value="Login" />
107 <span class="forgot"> Forgot your <a href="#" class="openUname">Username</a> or <a href=
"#" class="openPass">Password</a> ?</span></p>
108 </form>
109 </div>
110 </div>
111 </div>
```

Container with a Password reminder form.

Main content container

Menu container. Menu HTML content will be placed here after successfully login.

Content container. Control Panel HTML content will be placed here after successfully login.

If "authorized" session' variable is equal to "true" it means that we are logged in, so we have to control panel. The following code is the same as above one, but with additional styling

Login Form is hided now



```

112 <div class="forgotUname">
113     <p>Enter your registered e-mail and we'll send you your username</p>
114     <form method="post" action="javascript:retrieveUsername()" >
115         <span class="text-l"><span class="text-r">
116             <input tip="Only "demo@idangero.us" - will be the right e-mail" type="text" id=
"retrieveUnameEmail" size="20" />
117         </span></span>
118         <input style="margin-left:20px" type="submit" value="Send Username" />
119     </form>
120 </div>
121 <div class="forgotStatus"></div>
122 <div class="forgotPass">
123     <p>Enter your registered e-mail and we'll send you a NEW password</p>
124     <form method="post" action="javascript:retrievePassword()" >
125         <span class="text-l"><span class="text-r">
126             <input tip="Only "demo@idangero.us" - will be the right e-mail" type="text" id=
"retrievePassEmail" size="20" />
127         </span></span>
128         <input style="margin-left:20px" type="submit" value="Send Password" />
129     </form>
130 <div class="forgotStatus">Message with your Username was successfully sent</div>
131 </div>
132 <div class="clear"></div>
133 </div>
134 <div id="maincontent" style="display:block;height:100%;">
135     <div id="menu" style="height:100%">
136         <div class="menu" style="display:block"><?php include("includes/menu.php") ?></div>
137     </div>
138     <div class="content" style="display:block">
139         <div class="content-inner"><?php include("content/cp.php") ?></div>
140     </div>
141     <div class="clear"></div>
142 </div>
143 <script type="text/javascript">contentResize();menuFunctionality()</script>
144 <?php
145 }
146 ?>
147 <div class="dark-layer"></div>
148 <div id="popup">
149     
150     <div class="popup-content"></div>
151 </div>
152 <div id="tip">
153     <div class="tip-content"></div>
154     <div class="tip-arrow"></div>
155 </div>
156
157 </body>
158 </html>

```

Including menu content

Including Control Panel content

Also we have to call these two JavaScript functions to fix content size and to enable menu functionality

Popup window container

Tip container. Tip text will add here

## Logo

To change site **logo** you have to replace the logo.png (located in **images** folder) image to yours own or you could change the source of logo image to your image like this:

```
<div id="logo">  </div>
```



## Main Content Section

This section contains menu and content section for all content pages

All content pages (files) will be loaded into the div block with a “content-inner” class.

## Menu Section

Menu HTML content located in the **includes/menu.php** file:

```
1 <?php
2 session_start();
3 if(!(((($_SERVER['HTTP_X_REQUESTED_WITH'] == 'XMLHttpRequest')||defined('evolette'))
4  && $_SESSION['authorized']===true)) exit();
5 ?>
6 <ul>
7     <li class="menu-shortcuts"><a mtip="Control Panel" class="m-ajaxed" href=
8     "index.php?content=cp"></a><a mtip="Users" class="m-ajaxed" href=
10    "index.php?content=users"></a><a mtip="Send Message" class="m-popup" href=
12    "index.php?content=new-message"></a><a mtip="Site Settings" class="m-popup" href=
14    "index.php?content=settings"></a><a mtip="To Do" class="m-popup" href=
16    "index.php?content=todo"></a><a mtip="Preview Site" href="#" style="margin:0"><img src=
18    "images/menu-icons/preview.png" width="20" height="17" alt="Preview" /></a></li>
19    <li><span class="separator">Content</span>
20    <ul class="submenu">
21        <li><a class="m-ajaxed" href="index.php?content=articles">Articles</a>
22        <ul class="submenu">
23            <li><a class="m-ajaxed" href="index.php?content=article_add">Add New
24            Article</a></li>
25        </ul>
26        </li>
27        <li><a class="m-popup" href="index.php?content=categories">Categories</a></li>
28        <li><a class="m-popup" href="index.php?content=sections">Sections</a></li>
29    </ul>
30    </li>
31    <li><a class="m-popup" href="index.php?content=menu">Site Menu</a></li>
32    <li><a class="m-popup" href="index.php?content=modules">Modules</a></li>
33    <li><a class="m-popup" href="index.php?content=comments">Comments</a></li>
34    <li><a class="m-ajaxed" href="index.php?content=help">Evolette Features</a></li>
35 </ul>
```

Security code

Security code

Submenu

Submenu in submenu

To create your own menu link, first of all you have to create list element inside the parent **ul** list using the **<li>** tag. After that put the link inside using **<a>** and **</a>** tags.

To create submenu you have to create new **<ul>** list with a “**submenu**” class inside a parent **<li>** link.

## Popup Window

All content pages (files) will be loaded into the div block with a “popup-content” class.

## 7. Inner Linking

---

So let's see how the Inner Linking works. This you should to know first:

*All files with HTML content you want to load with Ajax must be located in the **content/** folder.*

*All files must have a **.php** extension.*

Evolette can handle links in three ways. All these ways are based on adding special CLASS attribute value to the links.

**Method 1.** You need to load file from the **content/** folder. For this case you have to add an “**ajaxed**” class (or “**m-ajaxed**” for menu links) for link, and to add filename as a value of content variable in the link HREF attribute. If you need to POST extra variables to this file you also have to specify them in the link HREF attribute. For example we need to load **articles.php** file with extra variables – **order\_by** and **filter\_by**. We have to use the following formatting:

ONLY filename, without extension

Additional variables will be sent (with POST method) to **articles.php** file. “**order\_by**” will be equal to “**id**” and “**filter\_by**” will be equal to “**category**”

```
<a class="ajaxed" href="index.php?content=articles&order_by=id&filter_by=category">Articles</a>  
or  
<a class="ajaxed" href="?content=articles&order_by=id&filter_by=category">Articles</a>
```

**Method 2.** You need to load file NOT from the **content/** folder. For this case you have to add an “**ajax-d**” (Ajax Direct) class (no equivalent for menu links) for link, and to specify direct URL for file in the link HREF attribute. If you need to post extra variables to this file you also have to specify them in the link HREF attribute. But in this case they will be posted with GET method. For example we need to load **articles.php** file from **myfolder/** folder with extra variables – **order\_by** and **filter\_by**. We have to use the following formatting:

Direct file location

Additional variables will be sent (with GET method) to **articles.php** file.  
“**order\_by**” will be equal to “**id**” and “**filter\_by**” will be equal to “**category**”

```
<a class="ajax-d" href="myfolder/articles.php?order_by=id&filter_by=category">Articles</a>
```

**Method 3.** You need ONLY to send variables to file from the **content/** folder. For this case you have to add a “**sendVars**” class (no equivalent for menu links) for link, and to add filename as a value of content variable in the link HREF attribute. All variables you need to send (with POST method) to this file you have to specify in the link HREF attribute. For example we need to send variables to **articles.php** file. These variables are **order\_by** and **filter\_by**. We have to use the following formatting:

ONLY filename, without extension

Additional variables will be sent (with POST method) to **articles.php** file.  
“**order\_by**” will be equal to “**id**” and “**filter\_by**” will be equal to “**category**”

```
<a class="sendVars" href="index.php?content=articles&order_by=id&filter_by=category">Send Vars</a>  
or  
<a class="sendVars" href="?content=articles&order_by=id&filter_by=category">Send Vars</a>
```

*But it is not recommended to use 3<sup>d</sup> method, because we will not see result of php script. It is much better to use method based on the Popup Window. Read more about this method in chapter 11.*

## 8. CSS Files and Structure

---

There is one CSS file used in this Template. You can find css attachment in the HEAD section of index.php file.

```
<head>  
-----  
<link rel="stylesheet" type="text/css" href="css/evolette.css" />  
-----  
</head>
```

By default it is located in the **css/** folder. You can easily change the path or filename in the **href** attribute of **link** tag.

**evolette.css** file contains all general layouts for the template. It is separated into sections:

```
/* --- Common Rules --- */
```

*There are common rules in this section, such as font size, fonts, link colors, layout images etc.*

```
/*----- Ajax Loader-----*/
```

*Ajax Loader styling*

```
/* --- Headings --- */
```

*Rules for headings.*

```
/* --- Logo--- */
```

*This section contains style rules for Logo*

```
/* --- Login Page--- */
```

*This section contains style rules for Login Page*

```
/* --- Login Form--- */
```

*This section contains style rules for Login Form*

```
/* --- Forgot Forms --- */
```

*This section contains style rules for reminder forms*

```
/* --- Menu --- */
```

*Rules for menu and navigation buttons*

```
/* --- Main Content--- */
```

*All styling rules for content pages*

```
/* --- Tip--- */
```

*This section contains style rules for TIP*

```
/* --- Control Panel--- */
```

*This section contains style rules for control panel icons*

```
/* --- Layouts--- */
```

*This section contains style rules for layouts*

```
/* --- Panes--- */
```

*This section contains style rules for panes*

```
/* --- Accordion--- */
```

*This section contains style rules for accordion*

```
/* --- Tabs--- */
```

*This section contains style rules for Tabs*

```
/* --- Tables--- */
```

*This section contains style rules for tables*

```
/* --- Popup Window--- */
```

*Styling rules for the popup window*

```
/* --- Form Buttons--- */
```

*This section contains style rules for buttons*

```
/* --- From--- */
```

*This section contains style rules for form*

```
/* --- Pagination--- */
```

*Pagination styling*

```
/* --- Typography--- */
```

*Notification styles*

```
/*-----End of CSS File-----*/
```

If you would like to edit a specific section of the site, simply find the appropriate section in the CSS file, and then scroll down until you find the appropriate style that needs to be edited.

## 9. Typography

---

To learn Evolette' typography on examples you have to go to **typography/** folder from the downloaded archive and open **index.html** file in your browser.

# 10. JavaScripts

---

Evolette uses three JavaScript files

Attachment of JavaScript files can be found in the HEAD section of index.php file:

```
<head>
-----
<script type="text/javascript" src="scripts/jquery-1.4.4.min.js"></script>
<script type="text/javascript" src="tiny_mce/jquery.tinymce.js"></script>
<script type="text/javascript" src="scripts/evolette.js"></script>
-----
</head>
```

You can easily change the path or filename in the **src** attribute of **script** tag.

The first script file is the jQuery (JavaScript library that greatly reduces the amount of code that you must write).

The second script file (jquery.tinymce.js) is the TinyMCE WYSIWYG text editor. You can learn more about TinyMCE editor [here](#).

And the last script file (evolette.js) is the custom script used for animation, effects and Ajax requests. You can learn more about jQuery methods used here (`fadeOut()`, `fadeIn()`, `fadeTo()`, `animate()`, `jQuery.get()`, `html()`, `delay()`, `hover()` etc.) by visiting this [jQuery Documentation](#) site.

Let's look at the custom script more closely.

The first part of the script is executed when the document is loaded. Here are the effects and rules for handling Login Form, its animation, Ajax loader, functions used for popup window and settings for TinyMCE editor



Settings for TinyMCE editor

```
1 // JavaScript Document
2 var editor_settings = {
3     script_url : "tiny_mce/tiny_mce.js",
4     theme : "advanced",
5     plugins :
6     "pagebreak,style,layer,table,save,advhr,advimage,advlink,emotions,iespell,inlinepopups,insertdate
7     time,preview,media,searchreplace,print,contextmenu,paste,directionality,fullscreen,noneditable,vi
8     sualchars,nonbreaking,xhtmlxtras,template,advlist",
9     theme_advanced_buttons1 :
10    "save,newdocument,|,bold,italic,underline,strikethrough,|,justifyleft,justifycenter,justifyright,
11    justifyfull,styleselect,formatselect,fontselect,fontsizeselect",
12    theme_advanced_buttons2 :
13    "cut,copy,paste,pastetext,pasteword,|,search,replace,|,bullist,numlist,|,outdent,indent,blockquot
14    e,|,undo,redo,|,link,unlink,anchor,image,cleanup,help,code,|,insertdate,inserttime,preview,|,fore
15    color,backcolor",
16    theme_advanced_buttons3 :
17    "tablecontrols,|,hr,removeformat,visualaid,|,sub,sup,|,charmap,emotions,iespell,media,advhr,|,pri
18    nt,|,ltr,rtl,|,fullscreen",
19    theme_advanced_buttons4 :
20    "insertlayer,movelforward,movelforward,absolute,|,styleprops,|,cite,abbr,acronym,del,ins,attribs,|
21    ,visualchars,nonbreaking",
22    theme_advanced_toolbar_location : "top",
23    theme_advanced_toolbar_align : "left",
24    theme_advanced_statusbar_location : "bottom",
25    theme_advanced_resizing : true,
26    template_external_list_url : "lists/template_list.js",
27    external_link_list_url : "lists/link_list.js",
28    external_image_list_url : "lists/image_list.js",
29    media_external_list_url : "lists/media_list.js"
30 }
31
32 $(function() {
33     mainFunctionality()
34     /*-----Login Form-----*/
35     $('#loginForm').submit(function(e) {
36         e.preventDefault();
37         var username = $('#username').val()
38         var password = $('#password').val()
39         jQuery.post("includes/login-check.php", {username:username,password:password}, function(
40         checkStatus) {
41             if (checkStatus == 1) $('#.wrong-status').fadeOut(300,function() {
42                 login()
43             })
44             else $('#.wrong-status').hide().fadeIn(300)
45         })
46     })
47     $('#a.openName').click(function(e) {
48         e.preventDefault();
49         $('#.forgotPass').fadeOut(200)
50         if($('#.forgotName').css('display')=='none') {
51             $('#.forgotName').slideDown(600)
52         }
53         else $('#.forgotName').fadeOut(200)
54     })
55     $('#a.openPass').click(function(e) {
56         e.preventDefault();
57         $('#.forgotName').fadeOut(200)
58         if($('#.forgotPass').css('display')=='none') {
59             $('#.forgotPass').slideDown(600)
60         }
61         else $('#.forgotPass').fadeOut(200)
62     })
63     /*----- Fix Content Size -----*/
64     $(window).resize(function() {
65         contentResize()
66     })
67 })
```

We have to call this function when the site is loaded.

Login Form handling. When the form is submitted, function sends username and password values with POST method to includes/login-check.php script. And if this script returns "1", then the **login()** function will be executed, otherwise the status message will be shown with fade animation for 300ms.

When the link with a "openName" class will be clicked, then the Password reminder form will be faded for 200ms,

If the Username reminder form is hidden, then it will be show with slide animation for 600ms

Otherwise it will be hided with fade animation for 200ms

The same code for Password reminder form

When the browser window is resized, then the **contentResize()** function will be executed

Slide effect for menu. After the window is scrolled, menu (ul) will slide to the scrolled distance for 600 ms after the 500ms delay.

```
53 /*----- Slide Menu -----*/
54 $(window).scroll(function() {
55     $('.menu > ul').stop(true, false).delay(500).animate({top:$(window).scrollTop()}, 600)
56 })
57 /*----- Logout -----*/
58 $("#logout").click(function() {logOut()})
59 /*----- Close Popup -----*/
60 $('.close-popup, .dark-layer').click(function() {closePopup()})
61 /*----- Ajax Loader -----*/
62 $("#ajax-loader").ajaxStart(function() {
63     $(this).css({display:"block"})
64 })
65 $("#ajax-loader").ajaxComplete(function() {
66     $(this).css({display:"none"});
67 })
68 })
```

When we'll click on the "logout" image (div with id="logout") the **logOut()** function will be executed

When we will click on the close image or on the "dark layer", then the **closePopup()** function will be executed

This code means that when the Ajax request is starting, the Ajax Loader will be shown.

When the Ajax request is completed, the Ajax Loader will be hidden

The next part of code contains functions which could be called from any place of the document. These functions are used to load files with content, for submitting and validation of forms, to call a popup window, to handle links and menu links, etc:

This is the main function. It is called every time when new page is loaded.

```
69 function mainFunctionality() {
70     $(function() {
71         /*----- Links Handling -----*/
72         $('a.ajaxed').click(function(e) {
73             e.preventDefault();
74             var url = $(this).attr('href')
75             loadContent('content/'+getVars(url, 'content')+'.php', getVars(url))
76         })
77         $('a.ajaxe-d').click(function(e) {
78             e.preventDefault();
79             loadContent($(this).attr('href'))
80         })
81         $('a.sendVars').click(function(e) {
82             e.preventDefault();
83             var url = $(this).attr('href')
84             sendVars('content/'+getVars(url, 'content')+'.php', getVars(url))
85         })
86         $('a.popup').click(function(e) {
87             e.preventDefault();
88             var url = $(this).attr('href')
89             var width = $(this).attr('rel');
90             var height = $(this).attr('rev');
91             popUp({url:'content/'+getVars(url, 'content')+'.php', width:width, height:height},
92                 getVars(url))
93         })
94         $('a.sendMessage').click(function(e) {
95             e.preventDefault();
96             popUp({url:"content/new-message.php"}, {mailto:$(this).attr('href')})
97         })
98     })
99 }
```

Handling for the links with an "ajaxed" class. When we will click on it, then the **loadContent()** function will be executed. URL for this case is the content/ folder plus filename as a value of content variable from href attribute plus .php extension. And additional variables from HREF attribute will be also sent.

Handling for the links with an "ajaxe-d" class. When we will click on it, then the **loadContent()** function will be executed. URL for this case is the full value of the HREF attribute.

Handling for the links with a "sendVars" class. When we will click on it, then the **sendVars()** function will be executed. URL and additional variables will be the same as for "ajaxed" class

Handling for the links with a "popup" class. When we will click on it, then the **popUp()** function will be executed. URL and additional variables will be the same as for "ajaxed" class. Width value will be equal to the REL attribute, and Height – REV attribute

Handling for the links with a "sendMessage" class. When we will click on it, then the **popUp()** function will be executed. New-message.php file will be placed to popup container with a filled "mailto" field.

Handling for the Forms with a “formToContent” class. When the form is submitted, then the **loadContent()** function will be executed. URL for this case will be equal to ACTION attribute of form, and values of all form fields will be posted as additional variables.

Handling for the Forms with a “formToPopup” class. When the form is submitted, then the **popUp()** function will be executed. URL for this case will be equal to ACTION attribute of form, and values of all form fields will be posted as additional variables.

Handling for the Forms with a “sendVars” class. When the form is submitted, then the **sendVars()** function will be executed. URL for this case will be equal to ACTION attribute of form, and values of all form fields will be posted as additional variables.

Animation of control panel icons. First function will be executed when you'll take mouse cursor on this icon. It will fade to opacity 0.5 for 200ms. Second function will be executed when you'll take mouse cursor away from icon. It will fade to opacity 1 for 200ms.

When textarea has a “tinymce” class then the TinyMCE will be enabled for this textarea.

Tip will be added to any table with an “expandable” class

When we'll click on the THEAD of table with an “expandable” class, function will check is this table is expanded or not.

If this table is not expanded, then it will be expanded with 600ms animation.

If this table is expanded, then it will be collapsed with 600ms animation.

When we'll take mouse cursor on any element with a TIP attribute, then the openTip() function will be executed.

When we'll take mouse cursor away from element with a TIP attribute, then tip will be faded out for 100ms.

Wrappers. Their usage makes the creation of some elements much faster

HTML content of every tab heading will wrapped with these span elements

Any element with a “button” class will be wrapped with these span elements:

```

97  /*----- Forms -----*/
98  $('form.formToContent').submit(function(e) {
99      e.preventDefault();
100     loadContent($(this).attr('action'),getFormVars($(this)))
101  })
102  $('form.formToPopup').submit(function(e) {
103      e.preventDefault();
104      popUp({url:$(this).attr('action')},getFormVars($(this)))
105  })
106  $('form.sendVars').submit(function(e) {
107      e.preventDefault();
108      sendVars($(this).attr('action'),getFormVars($(this)))
109  })
110  /*----- CP Icons -----*/
111  $('.cp-icons img').hover(
112      function() {
113          $(this).fadeTo(200,0.5)
114      },
115      function() {
116          $(this).fadeTo(200,1)
117      }
118  )
119  /*----- TinyMCE -----*/
120  $('textarea.tinymce').tinymce(editor_settings);
121  /*----- Expandable Tables -----*/
122  $('table.expandable thead').attr({'tip':'Click to expand table'})
123  $('table.expandable thead').click(function() {
124      var expandTable = $(this).closest('table');
125      if(!expandTable.hasClass('expanded')) {
126          var tableOffset = expandTable.offset().left-$('#menu').width();
127          expandTable.animate({width:'+='+(tableOffset*2-50),left:-tableOffset+25},600).
addClass('expanded');
128      }
129      else {
130          expandTable.animate({width:'100%',left:0},600).removeClass('expanded');
131      }
132  })
133  /*----- Tips -----*/
134  $('*[tip]').hover(
135      function() {
136          openTip($(this));
137      },
138      function() {
139          $('#tip').stop(true,true).fadeOut(100)
140      }
141  )
142  /*----- Wrappers -----*/
143  $('h3.pane-title').each(function() {
144      $(this).html('<span class="p-title-r"><span class="p-title-m">'+$(this).html()+
145  </span></span>')
146  })
147  $(".tabs-head li").each(function() {
148      $(this).html('<span class="p-title-r"><span class="p-title-m">'+$(this).html()+
149  </span></span>')
150  })
151  $(".button").wrap('<span class="button-l"><span class="button-m"></span></span>')
152  $(".w-button").wrap('<span class="w-button-l"><span class="w-button-m"></span></span>')
153  $(".bw-button").wrap('<span class="bw-button-l"><span class="bw-button-m"></span></span>')
154  $(".bb-button").wrap('<span class="bb-button-l"><span class="bb-button-m"></span></span>')

```

When we'll click on the accordion heading (element with an "acc-title" class, function will check is this heading is active or not. If accordion block has the "time-..." class, then animation time will be equal to this value, otherwise it will be equal to 300ms. Then all other accordion panes will be hidden with slide animation. And accordion pane which is following after the clicked accordion heading will be shown with slide animation

```

154  /*----- Accordion -----*/
155  $(".accordion .acc-title").click(function(){
156      if(!$(this).hasClass('current')) {
157          var accordion = $(this).closest('.accordion');
158          if (accordion.attr('class').indexOf('time-')>=0) var time = accordion.attr('class
159  ').split('time-')[1].split(' ')[0]*1;
160          else var time = 300;
161          $(this).next('.acc-pane').slideDown(time).nextAll('.acc-pane').slideUp(time).end
162  ().prevAll('.acc-pane').slideUp(time)
163          accordion.children('h3.current').removeClass('current')
164          $(this).addClass('current')
165      }
166  })
167  /*----- Tabs -----*/
168  $(".tabs-head li").click(function(){
169      if(!$(this).hasClass('current')) {
170          var tabs = $(this).closest('.tabs');
171          tabs.children('.tabs-pane').eq(tabs.find('li.current').index()).hide()
172          tabs.children('.tabs-pane').eq($(this).index()).show()
173          tabs.find('li.current').removeClass('current')
174          $(this).addClass('current')
175      }
176  })
177  function menuFunctionality(){
178      $(function(){
179          $('.submenu > li:last-child').css('border-bottom','none')
180          $('.menu a,.menu .separator').hover(
181              function(){
182                  $(this).fadeOut(150,0.5)
183              },
184              function(){
185                  $(this).fadeIn(150,1)
186              }
187          )
188          $('.menu li:has(ul.submenu)').hover(
189              function(){
190                  $(this).children('.submenu').stop(true,true).slideDown(300)
191              },
192              function(){
193                  $(this).children('.submenu').delay(300).slideUp(300)
194              }
195          )
196          /*----- Menu Links Handling -----*/
197          $('.a.m-ajaxed').click(function(e){
198              e.preventDefault();
199              var url = $(this).attr('href')
200              loadContent('content/'+getVars(url,'content')+'.php', getVars(url))
201          })
202          $('.a.m-popup').click(function(e){
203              e.preventDefault();
204              var url = $(this).attr('href')
205              popUp({url:'content/'+getVars(url,'content')+'.php'}, getVars(url))
206          })

```

Same rules for the Tabs

This function used to make all menu features available.

We have to remove border from the last link in submenus

Fade effect for menu links.

Submenu animation

The same as for the links with "ajaxed" class in mainFunctionality() function

The same as for the links with "popup" class in mainFunctionality() function



```

207  /*----- Menu Tip -----*/
208  $('*[mtip]').hover(
209      function() {
210          openTip($(this))
211      },
212      function() {
213          $('#tip').stop(true,true).fadeOut(100)
214      }
215  )
216  })
217  }
218  function contentResize() {
219      $('#maincontent').css({'height':$(document).height()})
220  }
221  function retrieveUsername() {
222      var email = $('#retrieveUnameEmail').val()
223      jQuery.post("includes/forgotUname.php", {email:email}, function(status) {
224          $(".forgotUname .forgotStatus").hide().html(status).fadeIn(300)
225      })
226  }
227  function retrievePassword() {
228      var email = $('#retrievePassEmail').val()
229      jQuery.post("includes/forgotPass.php", {email:email}, function(status) {
230          $(".forgotPass .forgotStatus").hide().html(status).fadeIn(300)
231      })
232  }
233  function login() {
234      jQuery.post("content/cp.php", function(controlPanel) {
235          jQuery.post("includes/menu.php", function(menu) {
236              $('.content-inner').html(controlPanel)
237              $('.menu').html(menu)
238              $('#login').fadeOut(600);
239              $('.lock').css({left:$('.lock').position().left}).delay(1200).animate({
240                  "left":0, "top":-5, "height":"-=1px", marginLeft:0}, 600)
241              $('#logo').css({left:$('#logo').position().left}).delay(1800).animate({
242                  "left":240, marginLeft:0}, 500)
243              $('#maincontent').delay(600).fadeIn(600)
244              $('#menu').delay(1800).animate({height:"100%"}, 1000)
245              $('.lock2').delay(3000).fadeOut(300, function() {
246                  $('.content').fadeIn(600, function() {
247                      contentResize()
248                      $('.menu').fadeIn(600)
249                  })
250                  mainFunctionality();
251                  menuFunctionality()
252                  $('#logout').delay(100).fadeIn(300)
253              })
254          })
255      })
256  }
257  function logout() {
258      jQuery.post('includes/logout.php');
259      $(".content").fadeOut(600, function() {contentResize(); $('.content-inner').html("");});
260      $('.menu').delay(300).fadeOut(600, function() {$('.menu').html("");});
261      $('#menu').delay(1000).animate({height:"0"}, 1000);
262      $('#logo').delay(2000).animate({left:"50%", marginLeft:"-350px"}, 600);
263      $('.lock').delay(2000).animate({left:"50%", marginLeft:"-350px", top:200, height:"+=1px"}, 600);
264      $('#maincontent').delay(2000).fadeOut(300);
265      $('#login').delay(2700).fadeIn(600);
266      $('#logout').delay(3300).fadeOut(300, function() {$('.lock2').delay(100).fadeIn(300)});
267  }

```

The same as for the elements with TIP attribute in **mainFunctionality()** function

This function is called every time when new page is loaded. We have to use it to fix "maincontent" block size to make menu full height.

This function will be executed when the "reminder username" form is submitted. It will send an Ajax POST request and email value to the **includes/forgotUname.php** file. Php script will check this email and return status message, which will be show with fade animation

Same rules for the "reminder password" form

First of all, the control panel (CP) and menu content will be loaded. After that, CP content will be insertend in the "content-inner" block and menu content – in the "menu" block. Then Login form will be faded for 600ms. "lock" and "logo" will change their positions. "Maincontent" will be faded in for 600ms after the 600ms delay.

Menu height will be animated to 100% value.

After the ".lock2" block is faded out, then the ".content" block will fade in for 600ms, and after that contentResize() function will be executed, and menu links will fade in for 600ms

These functions will be executed after the ".lock2" block is faded out

And "logout" block will fade in

We need to make Ajax request to this file to unset "authorized" session variable

The same rules as in Login function, but reversed. Also we have to remove "content-inner" and menu content.

This is one of the main functions. It provides opportunity to load any file with HTML content and additional variables to the "content" section of site.

First of all we have to close popup window

```
268 function loadContent(url,vars){
269     closePopup();
270     vars=vars||{};
271     jQuery.post(url,vars,function(newContent){
272         $('.content').fadeOut(600,function(){
273             $('.content-inner').html(newContent)
274             $('.content').fadeIn(600,function(){
275                 if($(window).scrollTop()>($('.content').height()+200)){
276                     if($('.html').css('scrollTop')==null){
277                         $('.body').delay(300).animate({scrollTop:150}, 900);
278                     }
279                     else {
280                         $('.html').delay(300).animate({scrollTop:150}, 900);
281                     }
282                 }
283             })
284             mainFunctionality();
285             contentResize()
286         })
287     })
288 }
289 function popUp(params,vars) {
290     params=params||{};
291     vars = vars || {};
292     params.width = params.width || 700;
293     params.height = params.height || 'auto';
294     if(params.url) {
295         jQuery.post(params.url,vars,function(data){
296             $('.popup-content').html(data)
297             popUp2();
298         })
299     }
300     else {
301         if(params.length) {
302             $('.popup-content').html(params);popUp2();
303         }
304         else {
305             $('.popup-content').html(params.text);popUp2()
306         }
307     }
308     function popUp2(){
309         if(params.width) $('.popup-content').css({width:params.width})
310         if(params.height) $('.popup-content').css({height:params.height})
311         $('#popup').css({marginLeft:-params.width/2-10})
312         $('.dark-layer').fadeOut(300,0.7);
313         $('#popup').css({top:$(window).scrollTop()+100})
314         $('#popup').delay(400).fadeIn(300)
315     }
316 }
317 function closePopup() {
318     $('.dark-layer,#popup').fadeOut(300,function(){
319         $('.popup-content').html("");
320     });
321 }
322 function sendVars(url,vars) {
323     jQuery.post(url,vars)
324 }
```

If additional variables are not specified then it will be equal to empty object

Ajax POST request to the file (specified with URL attribute) with additional variables

Then we have to hide content section with fade animation, and to put there new content loaded from the file

This case is for WEBKIT browsers

This for other browsers

If browser window scrolled down and new page is not visible, we have to scroll up browser window

Also we have to execute these two functions

popUp() function provides an opportunity to call a Popup Window. And to put inside the window any file with HTML content and additional variables. Or we can call a popup window with any plain text inside

If additional variables and parameters are not specified then they will be equal to empty object

Default width value (700px) – if width attribute is not specified

Default height value (auto) – if height attribute is not specified

If URL parameter is specified it means that we want to load a file. Then we have to send Ajax POST request to the specified file with additional variables. After that new content (from file) will be inserted into the div with a "popup-content" class and popUp2() function will be executed

This case used to put into popup window a plain text

This case used to put into popup window a plain text, specified in the text parameter

Setting up popup window sizes.

Setting up popup window margin-left position.

Then the "dark-layer" will be faded to opacity 0.7 for 300ms

After all popup window will be faded in for 300ms after the 400ms delay

Top position for popup window

closePopup() function used to hide popup window and "dark-layer" for 300ms. And after, the popup content will be removed

sendVars() function provides opportunity to send additional variables with POST method to specified file without any callback.



This function provides an opportunity to get variables from the string URL (or HREF attribute) of link. It works like a `parse_str()` PHP function.

First of all we have to split string with variables on two parts – string before “?” and string after “?” (we’ll need a second part)

```
325 function getVars(varString, only) {
326     varString.indexOf('?') >= 0 ? varString = varString.split("?")[1] : varString;
327     var pairs = varString.split('&');
328     var object1 = {};
329     for (var i = 0; i < pairs.length; i++) {
330         if (pairs[i].split('=')[0] == only) return pairs[i].split('=')[1].split('&')[0];
331         object1[pairs[i].split('=')[0]] = pairs[i].split('=')[1];
332     }
333     if (only) return null;
334     return object1;
335 }
```

Then we have to split this string to pairs like: “variable\_name=variable\_value” divided by “&” character

Object1 will contain all variables as its own properties. Then every pair we have to divide with “=” character. Object1 property name will be equal to variable name, object property value will be equal to variable value. After all, we have to return this object with all variables (properties) and their values

```
336 function getFormVars(form) {
337     var formVars = {};
338     var fieldName, fieldVal;
339     form.find('input, select, textarea').each(function() {
340         fieldName = $(this).attr('name');
341         if ($(this).attr('type') == "radio") {
342             if ($(this).is(':checked')) {
343                 fieldVal = $(this).val();
344                 formVars[fieldName] = fieldVal;
345             }
346         }
347         else {
348             if ($(this).attr('type') == "checkbox") {
349                 if (fieldName.indexOf('[]') > 0) {
350                     if (!formVars[fieldName]) formVars[fieldName] = [];
351                     if ($(this).is(':checked')) {
352                         fieldVal = $(this).val();
353                         formVars[fieldName].push(fieldVal);
354                     }
355                 }
356                 else {
357                     if ($(this).is(':checked')) {
358                         fieldVal = $(this).val();
359                         formVars[fieldName] = fieldVal;
360                     }
361                 }
362             }
363             else {
364                 fieldVal = $(this).val();
365                 formVars[fieldName] = fieldVal;
366             }
367         }
368     })
369     return formVars;
370 }
```

`getFormVars()` is the powerful function which takes all form fields values and returns them as an object with properties and values

First of all we have to find all inputs, selects and textareas in form. Fieldname will be equal to variable name.

If form field is “radio” we have to check is it checked or not. If it is checked then we’ll add new variable to formVars object

This case is for “checkbox” fields. First of all we have to check is it checkbox is array or not

If it is array, then we’ll add new array with “checked” values to formVars object

If it is not array, we’ll add new variable to formVars object if this “checkbox” is checked

This case for all other form fields (text inputs, buttons, selects and textareas)

After all, we have to return this object with all variables (properties) and their values

This function is used to call a tip window

We have to add to “tip-content” div block value of a TIP (or MTIP) attribute

Then we have to calculate “tip” position to set it above the element with TIP attribute

```
371 function openTip(e) {
372     $(".tip-content").html(e.attr('tip') || e.attr('mtip'))
373     var left = e.offset().left - ($("#tip").width() - e.width()) / 2
374     left < 0 ? left = 0 : left = left
375     var top = e.offset().top - ($("#tip").height() - 20)
376     top < 0 ? top = 0 : top = top
377     $("#tip").stop(true, true).css({"left": left, "top": top}).fadeIn(200)
378 }
```

Then we have to set “tip” position and to show it with fade animation

# 11. JavaScript Functions API

---

In this chapter we'll learn how to use main functions described in the previous chapter.

## *loadContent(url,vars) function*

As mentioned above this is one of the main functions. It provides opportunity to load any file with HTML content and additional variables to the “content” section of site.

Let's see how to use it:

`loadContent(url,vars)`

- `url` – string value. **Required**. Direct location of a file you want to load
- `vars` – JSON formatted variables and their values. Not required.

**Example 1.** We want to load **about.php** file (located in **content/** folder) without additional variables:

```
loadContent('content/about.php')
```

**Example 2.** We want to load **users.php** file (located in **content/** folder) with additional variables:

```
loadContent('content/users.php',{order_by:'id',filter_by:'online_status'})
```

## *popUp(params,vars) function*

This function provides an opportunity to call a Popup Window. And to put inside the window any file with HTML content and additional variables. Or we can call a popup window with any plain text inside.

Popup function can be called with two types of attributes:

## 1) `popUp(params, vars)`

- `params` – JSON formatted popup window parameters. **Required**. There are three acceptable parameters:
  - `url` - string value. **Required (only if TEXT parameter is not specified)**. Direct location of a file you want to load
  - `text` - string value. **Required (only if URL parameter is not specified)**. Plain text which you want to put inside the popup window
  - `width` – integer number. Not required. Value of the popup window width (default is 700)
  - `height` – integer number. Not required. Value of the popup window height (default is 'auto')
- `vars` – JSON formatted variables and their values. Not required.

## 2) `popUp(text)`

- `text` – string value. **Required**. Plain text which you want to put inside the popup window

**Example 1.** We want to load to popup window **about.php** file (located in **content/** folder) without additional variables, but we want to set popup window size:

```
popUp({url:'content/about.php',width:900,height:400})
```

**Example 2.** We want to load **users.php** file (located in **content/** folder) with additional variables, and we need to set popup window size:

```
popUp({url:'content/users.php',width:900,height:400},{order_by:'id',filter_by:'online_status'})
```

**Example 3.** The same as previous ones but with default window size:

```
popUp({url:'content/users.php'}, {order_by:'id',filter_by:'online_status'})
```

```
popUp({url:'content/about.php'})
```

**Example 4.** We want to put plain text into popup window, and we need to set popup window size:

```
popup({text:'This is a simple plain text',width:300,height:100})
```

But if you need to put plain text into popup window but without window size, it is much better to use simplified version of **popup()** function:

```
popup('This is a simple plain text')
```

## *getVarString(varString,only) function*

This function accepts URL string with variables and returns JSON formatted variables and their values.

Let's see how to use it:

```
getVarString(varString,only)
```

- `varString` – string value. **Required.** URL string with values.
- `only` – string value. Name of the variable whose value must be obtained. Not required.

Acceptable examples of `varString`:

- `index.php?content=articles&id=56&...`
- `http://www.mysite.com/index.php?content=articles&id=56&...`
- `?content=articles&id=56&...`
- `content=articles&id=56&...`

Now let's see what values will return this function:

```
getVarString("http://mysite.com/index.php?content=articles&order_by=id&filter_by=category")
```

will return

```
Object { content="articles", order_by="id", filter_by="category" }
```

```
getVarString("content=articles&order_by=id&filter_by=category")
```

will return

```
Object { content="articles", order_by="id", filter_by="category" }
```

```
getVars("content=articles")
```

will return



```
Object { content="articles" }
```

```
getVars("content=articles&order_by=id&filter_by=category", "content")
```

will return



```
:"articles"
```

ONLY variable is equal to "content" so will return string value of this variable.

## *getFormVars(form) function*

This function accepts form element and returns JSON formatted variables and their values of all form fields (inputs, selects, textareas).

Let's see how to use it:

```
getFormVars( form )
```

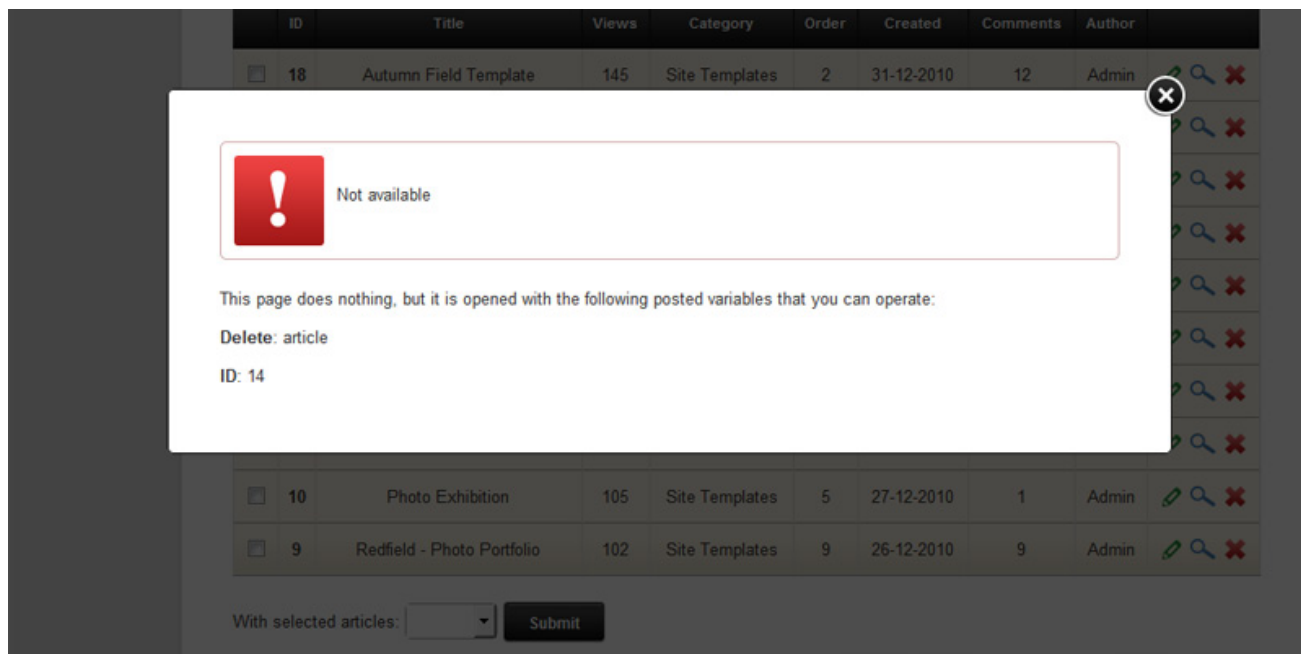
- `form` – string value. **Required**. You can use here jQuery' \$ wrapped selector.

For example if you have form with id attribute equal to "myform" you could call this function like this:

```
getFormVars( $( "#myform" ) )
```

## 12. Popup Window

**Evolette** template has its own engine to call popup windows. Here is screenshot:



Now let's see how the popup window works and how to call it with a desired content without using JavaScript `popup()` function directly:

**Example.** We need to call a popup window with HTML content from the *content/about.php* file. We can use the following formatting for the links:

Use "**popup**" (or "**m-popup**" for menu links) class to call a popup window with HTML content from file

Specify **filename** as a value of the **content** variable in the **href** attribute

Use **rel** attribute to specify window width in px (not required – 600px is default)

And **rev** attribute for window height in px (not required – 'auto' is default)

```
<a class="popup" href="index.php?content=about" rel="800" rev="200">About</a>
```

JavaScript automatically will include content from the **PHP** file which is located in the **content/** folder with filename specified in **content** variable (in this case is the **about**)



# 13. Forms Handling

---

So let's see how to handle forms.

Evolette can handle forms in three ways. All these ways are based on adding special CLASS attribute value to the form.

**Method 1.** You need to send values of Form fields to some file, and to load this file with posted variables into content section. For this case you have to add a “**formToContent**” class for form, and to specify file location in ACTION form attribute. For example we need to send form fields to the **content/articles.php** file and to load this file into content section We have to use the following formatting for FORM tag:

Diagram illustrating the formatting for Method 1. It shows a code snippet for a form tag with three callout boxes explaining its attributes:

- Direct file location**: Points to the `action="content/articles.php"` attribute.
- formToContent class will tell to JavaScript how to handle form**: Points to the `class="formToContent"` attribute.
- Doesn't matter, what method will be specified here. It always will be the POST**: Points to the `method="post"` attribute.

```
<form action="content/articles.php" class="formToContent" method="post">
..... Some Form Fields .....
</form>
```

**Method 2.** You need to send values of Form fields to some file, and to load this file with posted variables into popup window. For this case you have to add a “**formToPopup**” class for form, and to specify file location in ACTION form attribute. For example we need to send form fields to the **content/articles.php** file and to put this file inside the popup window. We have to use the following formatting for FORM tag:

Diagram illustrating the formatting for Method 2. It shows a code snippet for a form tag with three callout boxes explaining its attributes:

- Direct file location**: Points to the `action="content/articles.php"` attribute.
- formToPopup class will tell to JavaScript how to handle form**: Points to the `class="formToPopup"` attribute.
- Doesn't matter, what method will be specified here. It always will be the POST**: Points to the `method="get"` attribute.

```
<form action="content/articles.php" class="formToPopup" method="get">
..... Some Form Fields .....
</form>
```

**Method 3.** You ONLY need to send values of Form fields to some file, without any callback. For this case you have to add a “**sendVars**” class for form, and to specify file location in ACTION form attribute. For example we need to send form fields to the **content/articles.php** file. We have to use the following formatting for FORM tag:

Diagram illustrating the formatting for Method 3. It shows a code snippet for a form tag with three callout boxes explaining its attributes:

- Direct file location**: Points to the `action="content/articles.php"` attribute.
- sendVars class will tell to JavaScript how to handle form**: Points to the `class="sendVars"` attribute.
- Doesn't matter, what method will be specified here. It always will be the POST**: Points to the `method="get"` attribute.

```
<form action="content/articles.php" class="sendVars" method="get">
..... Some Form Fields .....
</form>
```

But it is not recommended to use 3<sup>rd</sup> method, because we will not see result of php script. It is much better to use method based on the Popup Window.

## 14. Mailing Engine

Evolette has its own simple mailing engine. Actually it is a simple php file with mailing form, which should be opened in popup window. It is located in the **content/new-message.php** file:

Security code

```
1 <?php
2 session_start();
3 if(!(((($_SERVER['HTTP_X_REQUESTED_WITH'] == 'XMLHttpRequest')||defined('evolette')) && $_SESSION
  ['authorized']==true)) exit();
4 function sendMail($mailto,$mailFrom,$subject,$message) {
5     $headers = "Date: ".date("j M Y H:i:s")."\r\n";
6     $headers .= "From: $mailFrom\r\n";
7     $headers .= "Reply-To: $mailFrom\r\n";
8     $headers .= "Return-Path: $mailFrom\r\n";
9     $headers .= "X-Mailer: PHPMailer". "\r\n";
10    $headers .= "Content-Transfer-Encoding: 8bit\r\n";
11    $headers .= "Content-Type: text/plain; charset=\\"utf-8\\"";
12    if(mail($mailto, $subject, $message, $headers)) {
13        return true;
14    }
15    else return false;
16 }
17 if(!empty($_POST['sendMessage'])) {
18     if(sendMail($_POST['mailto'],$_POST['mailfrom'],$_POST['subject'],$_POST['message'])) {
19         $status = '<div class="approve"><p>Message successfully sent!</p></div>';
20     }
21     else {
22         $status = '<div class="alert"><p>Error Occured!</p></div>';
23     }
24 }
25
26 ?>
27 <script type="text/javascript">
28 $(function(){
29     $('#messageStatus').slideDown(600).delay(3000).slideUp(600)
30     $('#usersEmails').change(function(){
31         $('#input[name=mailto]').val($(this).val())
32     })
33 })
34 </script>
35
36 <h2 class="content-heading h-mail">New Message:</h2>
37 <div id="messageStatus" style="display:none;"><? echo $status ?></div>
38 <form onsubmit="popUp({url:'content/new-message.php'},getFormVars($(this)));return false" method
  ="post" action="content/new-message.php">
39     <p>
40         <label><strong>From:</strong><br />
41         <input style="color:#777" class="text" type="text" name="mailfrom" readonly="readonly"
  value="Evolette Administrator <demo@idangero.us>" size="40" />
42     </label>
43     </p>
```

sendMail() php function used to send e-mail messages. It almost the same as the default mail() function, but it comes with more extended headers information. This function returns "true" if the message is sent, otherwise it returns "false".

If "sendMessage" button is clicked, script will send an e-mail message, and if the message is sent \$status variable will contain successfully message, otherwise it will contain message with error.

We have to show status message. It will be show with slide animation for 600ms. Then delay for 3seconds, and it will be hidden

When we'll select some user from "Users" select field, it will be inserted to input field with a "mailto" name attribute.

Status message will be inserted here

```

44 <p>
45   <label><strong>To:</strong><br />
46   <input class="text" type="text" name="mailto" size="40" value="<? echo $_POST['mailto'] ?>
" />
47   </label>
48   or choose from Users:
49   <select size="1" class="select" id="usersEmails">
50     <option></option>
51     <option value="demo@idangero.us">Michael Howard &lt;demo@idangero.us&gt;</option>
52     <option value="info@idangero.us">John Abramson &lt;info@idangero.us&gt;</option>
53     <option value="support@idangero.us">Ryan Attwood &lt;support@idangero.us&gt;</option>
54   </select>
55 </p>
56 <p>
57   <label><strong>Subject:</strong><br />
58   <input class="text" value="<? echo $_POST['subject'] ?>" type="text" name="subject" size="
40" />
59   </label>
60 </p>
61 <p>
62   <label><strong>Message:</strong><br />
63   <textarea name="message" class="textarea" style="width:690px" rows="12"><? echo $_POST[
'message'] ?></textarea>
64   </label>
65 </p>
66 <p><span class="bb-button-l"><span class="bb-button-m">
67   <input class="bb-button" type="submit" name="sendMessage" value="Send Message" />
68   </span></span></p>
69 </form>

```

Here is screenshot:

**New Message:**

From:  
Evolette Administrator <demo@idangero.us>

To:  
 or choose from Users:

Subject:

Message:

Now let's see how to open this New Message window.

1. It could be opened empty by a simple link with a "popup" class:

```
<a class="popup" href="index.php?content=new-message">New Message</a>
```

2. It could be opened with prefilled email address. In this case you have to use "sendMessage" class for link and to specify email address in the link' HREF attribute:

```
<a class="sendMessage" href="demo@idangero.us">Send Message to ID</a>
```

3. It could be opened with all prefilled fields. This case is the same as the first one, but you have to specify values of every field in the link' HREF attribute:

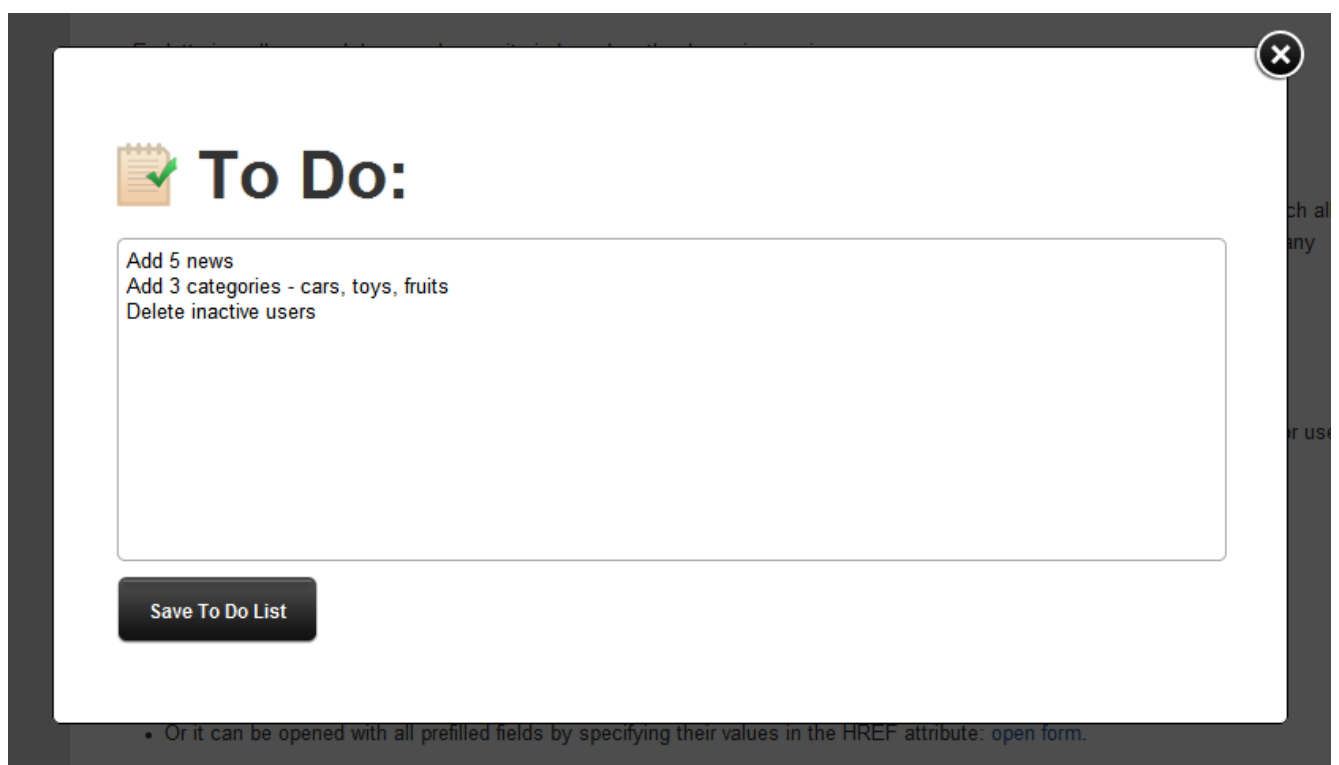
mailto value      subject value      message value

```
<a class="popup" href="content=new-messages&mailto=demo@idangero.us&subject=Demo Subject&message=This message posted from link"> Open with prefilled fields </a>
```

## 15. To Do List

Evolette has an editable To Do list. In this template To Do content is stored in the file. But if your CMS will use not one administrator user, it is better to save it to database for each user.

Here is screenshot:



Actually it is a simple php file with a form, which should be opened in popup window. It is located in the **content/todo.php** file. To Do content located in the **content/todo-txt.php** file.

```

1  <?php
2  session_start();
3  if(!((($_SERVER['HTTP_X_REQUESTED_WITH'] == 'XMLHttpRequest')||defined('evolette')) &&
4  $_SESSION['authorized']===true)) exit();
5  ?>
6  <?php
7  define ('todo',1);
8  if(!empty($_POST['updateToDo'])) {
9      $newToDo = "<?php if (!defined('todo')) exit();?>\r\n$_POST[todo]";
10
11     $fp = fopen('todo-txt.php', 'w');
12     if(fwrite ($fp,$newToDo)) $status = '<div class="approve"><p>To Do list has been
13     successfully saved!</p></div>';
14     else $status = '<div class="alert"><p>Error occurred!</p></div>';
15     fclose($fp);
16 }
17 ?>
18 <script type="text/javascript">
19 $(function(){
20     $('#ToDoStatus').slideDown(600).delay(3000).slideUp(600)
21 })
22 </script>
23 <h2 class="content-heading h-todo">To Do:</h2>
24 <div id="ToDoStatus" style="display:none;"><? echo $status ?></div>
25 <form onsubmit="popUp({url:'content/todo.php'},getFormVars($(this)));return false" method=
26 "post" action="content/todo.php">
27     <p>
28         <textarea name="todo" class="textarea" style="width:690px" rows="12"><?php include(
29         'todo-txt.php')?>
30     </textarea>
31     <p><span class="bb-button-1"><span class="bb-button-m">
32         <input class="bb-button" type="submit" name="updateToDo" value="Save To Do List" />
33     </span></span></p>
34 </form>

```

Security code

We need to define **todo** variable to make possible todo-txt.php file including.

If "updaToDo" button is clicked, script will add new ToDo file text into the **\$newToDo** variable. And then script will rewrite content of **todo-txt.php** file.

We have to show status message. It will be show with slide animation for 600ms. Then delay for 3seconds, and it will be hidden

If todo-txt.php file is successfully overwritten \$status variable will contain successfully message, otherwise it will contain message with error.

Status message will be inserted here

## 16. PSD Files & Structure

---

There are the **five** layered **.psd** files comes with the **Evolette** template. You can locate them into the **psd/** folder from the downloaded archive. Using these files you can easily redesign template. To edit these files you'll need an Adobe™ Photoshop™ (Recommended is CS4 or CS5 version)

- **login.psd**. This file contains login form
- **evolette-inside.psd**. All Evolette typography
- **big-icons.psd**. Icons used in Control Panel
- **small-icons.psd**. Small icons
- **ajax-loader.psd**. This file will help you to redesign Ajax loader

All layers in these file are well organized and easy to understand and redesign.

## 17. Licensing Terms

---

### Regular License (RL)

RL gives you as a customer non-exclusive & non-transferable right to use the product you've bought, in this case is the **Evolette** (further "**Item**")

1. We do not limit the number of **Item**'s copies you are going to use. Using one **Item** you can create, for example, even 100 web-sites.
2. You can use the **Item** by itself or it's also possible to apply it in other project you work at.
3. You can use the **Item** for your own purpose as well as for your clients.
4. You can use the **Item** in commercial projects.
5. You can reproduce the **Item**:
  - on a web-site or as a web-site
  - as a part of software you create not for sale
  - as a printed variant



- in digital(electronic) format (as a presentation or an e-book)
  - in video products
6. All photos used on iDangero.us for demonstration photo gallery functioning are our private property (if only there is nothing said about contrary). It's prohibited to use these photos wherever.
  7. You are not allowed to sell, resell, license or give the **Item** free (any way) without our written consent. Please, do not offer to do it to any person.
  8. You also do not have the right to use the **Item** in a project for selling (for example, for other templates, scripts, graphics and so on).
  9. It's prohibited to rework / redesign / reproduce the **Item** (i.e.to rename it or change graphics & so on) and after this to sell it as your own.
  - 10.In spite of reselling limitation you could claim money for the **Item** from your client.
  - 11.If the **Item** (the whole **Item** or its parts) is created with materials used by GNU General Public License (GPL) (or some other license) it means you should follow all the terms of the license using the **Item**.